MOTION TOMOGRAPHY PERFORMED BY UNDERWATER MOBILE SENSOR NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Dongsik Chang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2016

MOTION TOMOGRAPHY PERFORMED BY UNDERWATER MOBILE SENSOR NETWORKS

Approved by:

Professor Fumin Zhang, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Patricio A. Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Catherine R. Edwards
Skidaway Institute of Oceanography
University of Georgia

Professor WenZhan Song
Department of Computer Science
Georgia State University

Professor Ayanna M. Howard
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yang Wang
School of Civil and Environmental Engineering
Georgia Institute of Technology

Date Approved: 12 July 2016

To my father,

my mother,

and my brother.

ACKNOWLEDGEMENTS

This thesis would have not been possible without the help and support of many people during my Ph.D. study at the Georgia Institute of Technology. First and foremost, I would like to gratefully and sincerely thank my research advisor Dr. Fumin Zhang for his guidance, patience, and encouragements throughout my journey. Especially, his persistent guidance improved my analytical thinking and problem solving capabilities, and his passion and enthusiasm towards learning inspired me.

I would also like to thank my co-advisor Dr. Catherine R. Edwards. Her input for my research problems from the physical oceanography point of view helped me improve my capabilities of analyzing practical intuitions coupled with theoretical findings in engineering systems for oceanographic applications. Discussion with her on my research problems had been always pleasant.

I thank Dr. Ayanna M. Howard, Dr. Patricio A. Vela, Dr. Yang Wang, and Dr. WenZhan Song for being part of my Ph.D. dissertation committee and spending their valuable time reading my thesis and providing constructive comments where needed.

I thank my fellow lab members in the Georgia Tech Systems Research (GTSR) lab directed by Dr. Fumin Zhang for supporting my Ph.D. study and making my time in graduate school enjoyable. I have learned a lot by interacting with the GTSR members. I am also thankful to all my collaborators at the Skidaway Institute of Oceanography and the University of North Carolina at Chapel Hill during my Long Bay field experiments in 2012 and 2013.

Finally, I would like to thank my family and my friends. Without their love and support, this work would have not been accomplished.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The motion of underwater mobile sensing agents (UMSAs) is affected by ambient flow; knowledge of an environmental flow field can be used to improve the navigation of UMSAs. However, various flow data including existing regional ocean model output data often provide insufficient spatial and temporal resolutions for precise guidance of mobile sensing platforms. Due to imperfect knowledge of flow, the actual trajectory of a UMSA deviates from the predicted trajectory, which provides Lagrangian information of the underlying flow field.

The primary contribution of this thesis is a formulation of the problem of estimating an underlying flow field from Lagrangian trajectory information of UMSAs. The influence of an underlying flow field on the trajectory of a UMSA can be modeled as a line integral, formulating an inverse problem to infer an underlying flow field from the trajectory information of a UMSA. To numerically solve this inverse problem, this line integral is then discretized into algebraic matrix equations that are expressed as a nonlinear system of equations because of the dependency of the trajectories of UMSAs on a flow field. The *motion tomography* (MT) method presented in this thesis estimates an underlying flow field computationally faster and in higher resolution than existing flow models by solving an inverse problem for these algebraic equations.

Flow field mapping through MT is achieved by iterating two alternating key steps: *trajectory tracing* and *flow field estimation*. Trajectory tracing estimates unknown nonlinear vehicle trajectories and its error bound is determined by the error in estimated heading and that in estimated flow field. For flow field estimation, iterative sub-optimization methods referred to as *Kaczmarz-type methods* that solve a nonlinear system of equations constructed for MT are developed by extending from the

Kaczmarz method, which is well-known for computerized tomography in medical imaging. Then, based on these Kaczmarz-type methods, flow field estimation algorithms are developed. The MT method is first analyzed for time-invariant flow. Then, MT is modified to incorporate a data-driven flow model that approximates the flow using basis functions and corresponding parameters, formulating parametric MT. The convergence for the Kaczmarz-type methods developed for flow field estimation through non-parametric and parametric MT is analyzed. The validation and effectiveness of both MT schemes are demonstrated using simulation and experimental data.

The MT method is then extended to time-varying flow fields by two approaches. We first discretize an MT mapping domain in both space and time. Because of this spatio-temporal discretization, the solution variable may have a very high dimension and may be sparse (i.e., the solution may contain many zeros). To resolve this issue, a data-driven flow model with basis functions and corresponding parameters is employed, and spatial and temporal parameters are separately and iteratively estimated through MT to construct a time-varying flow field. Another approach uses a *generic environmental model* (GEM) that combines a data-driven flow model with real-time data streams collected from UMSAs. A constructed GEM assimilates both Eulerian data collected from moorings and Lagrangian data collected from UMSAs to resolve the lack of temporal variability in MT. Due to the coupling between temporal and spatial components in the flow model, estimation of temporal and spatial parameters for the GEM becomes a nonlinear filtering problem. By decomposing this nonlinear filtering problem into two linear sub-filtering problems, the developed method estimates both temporal and spatial parameters. Each of the sub-filters designed for temporal and spatial parameter estimation assimilates Eulerian data and Lagrangian data, respectively. To ensure that the fusion of both Eulerian and Lagrangian data streams resolves the lack of the temporal variability in MT, observability is analyzed.

Both approaches are demonstrated through simulations.

To obtain the best possible results, flow field estimation algorithms based on the Kaczmarz-type methods developed for MT should use collective trajectory information available from all the UMSAs. However, because of the limited communication capabilities of the vehicles or the large scale of applications, data may be collected or shared among vehicles in a distributed manner. In this case, only partial information may be available for MT. To account for such case, distributed MT is developed so that individual vehicles can estimate flow model parameters with knowledge of estimated parameters shared among their neighbors only. As part of the effort to develop distributed MT, a distributed Kaczmarz method for solving a linear system of equations is first developed. Then, it is extended to a distributed Kaczmarz-type method for solving a nonlinear system of equations formulated for MT. The convergence for both methods are analyzed, and the distributed Kaczmarz-type method is implemented for flow field estimation by distributed MT, which is validated through simulations.

CHAPTER I

INTRODUCTION

A group of underwater mobile sensing agents (UMSAs) that collectively gathers information about oceanic environments and adapts the behaviors of its members to environmental changes based on this information is referred to as an *underwater mobile sensor network* (UMSN). Because of the adaptive mobility and distributed sensing capability of its members, the UMSN has gained increased attention in oceanic environmental monitoring and data collection. Given a specific environmental data sampling pattern, the quality of data collected by a UMSA largely relies on its navigation performance. Since the motion of a UMSA is often affected by ambient flow along its trajectory, the importance of knowledge of flow for navigation of UMSAs has been increasingly emphasized.

During the navigation of UMSAs, the primary method for localization of UMSAs is the global positioning system (GPS), which provides position information of UMSAs on the surface of water. Since GPS signals cannot propagate through seawater, UMSAs typically estimate and predict their underwater positions based on a priori knowledge of the flow field. Ocean flow modeling is an active research area with significant applications in various areas such as marine robotics and oceanographic, atmospheric, and climatological forecasting. However, the high complexity and large scale of the ocean and the lack of direct measurements bring great challenges in ocean flow modeling and forecasting with a spatio-temporal resolution and a computational speed that are sufficient for the guidance of UMSAs. Due to imperfect knowledge of flow, the actual trajectory of a UMSA deviates from the predicted trajectory, providing Lagrangian information of the underlying flow field.

This dissertation introduces a novel method for modeling and estimating a flow field from the Lagrangian trajectory information of UMSAs computationally faster and in higher resolution than existing flow models. Considering that the influence of an underlying flow field on the motion of UMSAs accumulates along the trajectories of UMSAs, an inverse problem can be formulated to infer an underlying flow field from the trajectory information of UMSAs. The relationship between the flow field and the trajectory of a UMSA is first modeled as a line integral that is then discretized into algebraic matrix equations. Because of the dependency of the trajectories of UMSAs on a flow field, these algebraic equations are characterized as a nonlinear system of equations. Through solving this system of equations, we estimate a high-resolution map of a flow field. We refer to this method as *motion tomography* (MT).

We first formulate MT for a time-invariant flow field. Because the trajectories of UMSAs are typically unknown and nonlinear, MT constructs the system of equations using an initial guess of an underlying flow field and that of UMSA trajectories. Then, MT estimates a flow field map through an iterative process that alternates the following two key steps: 1) *trajectory tracing*, which estimates the trajectories of UMSAs using the latest flow field estimate and 2) *flow field estimation*, which estimates a flow field by solving the system of equations for MT through an inverse problem. By extending the Kaczmarz method which iteratively solves a sub-optimization problem for a linear system of equations, we develop a Kaczmarz-type method that solves the nonlinear system of equations constructed for MT without computing the gradient of the system of equations. We analyze the error bound for trajectory tracing and the convergence for the Kaczmarz-type method used for flow field estimation.

The system of equations that MT solves is typically underdetermined (i.e., the number of equations is fewer than that of unknowns). By employing a data-driven flow model that approximates the flow using basis functions with corresponding parameters, the unknowns of the system of equations becomes parameters of the model,

formulating parametric MT. With an appropriate number of basis functions, the dimension of the solution space may decrease, which may change the system of equations to overdetermined. In addition, if a flow field is smooth, the parameterization may reduce noise of the estimated flow field that may arise from uncertainties such as model error or measurement noise. We use a Kaczmarz-type method for MT to solve the nonlinear system of equations constructed for parametric MT and create a map of a flow field by using the estimated parameters.

To account for the coupling between temporal and spatial variations in flow modeling, we extend MT to a time-varying flow field by using two approaches. The first approach to incorporate temporal variability into MT formulates an inverse problem associated with the influence of time-varying flow on the UMSA trajectory in a spatially and temporally discretized domain. By solving this inverse problem, we obtain an estimate of an underlying time-varying flow field. Since the solution variable contains flow estimates for a spatially and temporally discretized domain, the dimension of the solution space may become very high. In addition, compared to the entire spatio-temporal domain, if each spatio-temporal subinterval is small, the solution variable may be sparse (i.e., it may contain many zeros). To deal with these issues, we incorporate a data-driven flow model. The MT method is then used to estimate parameters of the flow model.

Another approach employs stationary sensors such as a mooring providing Eulerian data, in addition to UMSAs collecting Lagrangian data. These two complementary types of data streams are assimilated into a generic environmental model (GEM) that is a data-driven flow model combined with real-time data streams collected from UMSAs. However, the coupling between temporal and spatial components in the flow model makes estimation of temporal and spatial parameters for the GEM a nonlinear filtering problem. We decompose this nonlinear filtering problem into two linear sub-filtering problems and design two sub-filters for temporal and spatial parameter

estimation that assimilate Eulerian data and Lagrangian data, respectively. The Lagrangian data have much slower time scale than the Eulerian data, but they provide necessary spatial coverage and spatial resolution. Therefore, by fusing these two types of data streams, they complement each other. To ensure that these two types of data streams construct a time-varying flow field, observability for a flow field is analyzed.

The Kaczmarz-type methods for MT flow field estimation require collective data from all the sensing agents in a UMSN, limiting its implementation to central computation, which may not be always possible. To consider the case in which the trajectory information of multiple vehicles cannot be obtained collectively, distributed implementation of the MT method, referred to as distributed MT, is developed. We first extend the Kaczmarz method to a distributed version. Then, we present a distributed version of the Kaczmarz-type method for solving the nonlinear system of equations constructed for MT. Extended from the Kaczmarz method, the both distributed methods solve iterative sub-optimization problems with partially shared knowledge of estimates in a networked system. In other words, by running the methods, each node shares its estimate of a solution to a system of equations at each iteration with other nodes nearby and computes a new estimate of the solution with knowledge of estimates shared by other nodes. We analyze the convergence for both distributed methods. The distributed method developed for MT is applied to distributed MT.

The rest of the document is organized as follows. Chapter 2 introduces motivation and background for this research work. Chapters 3 and 4 present the motion tomography method and its parameterized version. Chapter 5 extends motion tomography to a time-varying flow field, and Chapter 6 presents a distributed implementation of motion tomography. Finally, Chapter 7 concludes this document.

CHAPTER II

BACKGROUND AND MOTIVATION

This section reviews the recent development and techniques for UMSAs and introduces a motion model for UMSAs under flow. Then, we present a data-driven flow model and flow data sources considered in this document. Lastly, the description of the Long Bay deployments, which motivated the author's research on flow field modeling and estimation using vehicle trajectories, is provided.

2.1 Underwater Mobile Sensing Agents

In recent years, the use of a group of UMSAs, often called marine vehicles, for data collection in a large-scale dynamic oceanic environment has gained increased attention. One of the most exciting challenges for field robotics and control is to collect oceanic environmental data using UMSAs in order to enhance our knowledge of the ocean so that we understand physical, chemical, and biological processes in there more thoroughly [23]. Applications of great interest include the monitoring and data collection of environments such as chemical and biological plume mapping/tracking (e.g., oil and phytoplankton concentrations [60, 77]) and physical process monitoring (e.g., temperature and salinity [79]).

One type of UMSAs, autonomous underwater vehicles (AUVs), has emerged as a highly versatile technology for oceanic environmental sampling and monitoring [20, 28, 50]. They are flexible and integrated sensor platforms that can be roughly categorized into two groups: actively-propelled vehicles and gliders [63]. Compared to actively-propelled vehicles, gliders are characterized by low speed and long endurance because of their energy efficiency. UMSAs primarily use GPS for localization [56], but since GPS signals are not available underwater, they have limited localization capabilities

[91]. An overview of underwater vehicles is given in [90].

To effectively navigate UMSAs in an environment in the presence of flow, researchers have recently drawn their attention to the control and path planning of UMSAs under flow. Optimal path planning in fast-flowing estuarine environments is studied in [46], and optimal path planning under strong, time-varying currents is studied in [87], in which relevant experimental results are demonstrated. In [80], an optimal path planning algorithm that guarantees the existence of a path in strong currents is presented. A path planning algorithm in [25] adapts A*, a graph-based optimal path planning algorithm [34], for constant surfacing times of gliders and incorporates predictions from an ocean model. The use of predictive ocean models in path planning is more thoroughly studied in [78, 85, 68, 24, 12].

2.1.1 Horizontal Motion of UMSAs Under Flow

To model the influence of flow on the vehicle trajectory, we consider the horizontal motion of UMSAs in the presence of flow. The horizontal vehicle motion is modeled using a first-order particle model with constant speed, s_{h}, and vehicle heading, θ. Let us denote the time by $t \in \mathbb{R}$ and the position by $\mathbf{r} \in \mathbb{R}^2$. Let $\mathcal{T} = [t^0, t^f]$, with observation time horizon $T_{\mathrm{obs}} = t^f - t^0 > 0$, be a bounded time interval called observation interval. Suppose during the observation interval, the actual position, $\mathbf{r}$, of a vehicle is available only at $t = t^0$ and $t = t^f$ and a vehicle travels using a constant flow estimate or prediction, $\tilde{\mathbf{f}}$.

The predicted vehicle position, $\tilde{\mathbf{r}}(t)$, during the observation interval can be obtained by integrating

$$\dot{\tilde{\mathbf{r}}}(t) = s_{\mathrm{h}} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + \tilde{\mathbf{f}}. \tag{1}$$

Let us assume that vehicle heading θ is known. The actual vehicle position, $\mathbf{r}(t)$, is

affected by the real flow, $\mathbf{f}(\mathbf{r}, t)$, and can be obtained by

$$\begin{aligned} \dot{\mathbf{r}}(t) &= s_{\mathrm{h}} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + \mathbf{f}(\mathbf{r}, t) \\ &= \dot{\tilde{\mathbf{r}}}(t) + \mathbf{f}(\mathbf{r}, t) - \tilde{\mathbf{f}}, \end{aligned} \tag{2}$$

in which the second equation is obtained by substituting Equation (1) into Equation (2). Since real flow $\mathbf{f}$ is unknown, $\mathbf{r}(t)$ is unknown during $t \in (t^0, t^f)$. In contrast, estimated or predicted flow $\tilde{\mathbf{f}}$ is known along the vehicle trajectory; without loss of generality, we let $\tilde{\mathbf{f}} = \mathbf{0}$ for simplicity throughout the document. However, the results presented in this document apply to the general case when $\tilde{\mathbf{f}} \neq \mathbf{0}$.

Because of unknown flow $\mathbf{f}$, the actual trajectory differs from the predicted trajectory. This difference is referred to as the *motion-integration error*. Let $\gamma = \{\mathbf{r}(t)\}_{t \in \mathcal{T}} : \mathcal{T} \to \mathbb{R}^2$ be the trajectory of a vehicle. From Equations (1) and (2), the motion-integration error over one observation interval is given by

$$\begin{aligned} \mathbf{d}(\gamma, \mathcal{T}) &= \int_{t^0}^{t^f} \left(\dot{\mathbf{r}}(\tau) - \dot{\tilde{\mathbf{r}}}(\tau) \right) d\tau \\ &= \int_{t^0}^{t^f} \mathbf{f}(\mathbf{r}, \tau) d\tau. \end{aligned} \tag{3}$$

The motion-integration error has been used to estimate a flow field in the vicinity of an AUV that does not measure ambient flow directly. For example, the underwater glider, a buoyancy-driven AUV [21], compute a spatially and temporally averaged flow estimate from the motion-integration error along the vehicle trajectory traveled over one subsurface interval [57]. This method is very efficient in computation, and the glider incorporates this estimate into navigation to reduce the motion-integration error for the next subsurface interval. A similar way to estimate a flow velocity is presented in [69], in which the estimated flow is used to identify model parameters for a time-invariant flow field. However, the effectiveness of this method significantly degrades in the presence of flow with strong spatial and temporal variations [12].

Assuming these variations are dominated by tidal currents, the author's previous work [12] improved this method by incorporating predictions from a tidal model and validated the method in field experiments off the coast of Long Bay, South Carolina. This work inspired further study for flow field modeling and estimation based on the motion-integration error.

2.2 Data-Driven Flow Field Modeling

Flow field modeling and estimation have been studied in oceanography, fluid dynamics, and marine robotics. For geophysical flow, physics-based models numerically solve partial differential equations (PDEs) under known initial and boundary conditions. However, regional numerical models (e.g., [52, 4, 75, 33]) typically are computationally expensive and formulated with the spatial resolution on the order of kilometers and the temporal resolution on the order of hours to days, which may be considered as low resolution for the guidance of UMSAs. To obtain flow estimates computationally faster and in higher resolution than PDE-based models, data-driven models are introduced. For example, [59] and [7] designed flow models by using temporal and spatial decomposition techniques. In the following sections, we present a generalized form of such data-driven flow models. Then, we introduce two representations of a flow field: Eulerian and Lagrangian, and present typical Eulerian and Lagrangian data sources that can be used for data-driven flow models.

2.2.1 A Data-Driven Flow Model

Let us denote the time by $t \in \mathbb{R}$ and the position by $\mathbf{r} \in \mathbb{R}^2$. For position $\mathbf{r}$, we define a series of spatial basis functions indexed by m as $\phi_m(\mathbf{r}) \in \mathbb{R}$ to approximate spatial variations of flow. For time t, a series of temporal basis functions indexed by n is given by $\psi_n(t) \in \mathbb{R}$ which approximates temporal variations of flow. Let us define constant parameter vectors coupled with both spatial and temporal basis functions by $\eta_{(m,n)} \in \mathbb{R}^2$. Then, with M spatial basis functions and N temporal basis functions,

flow velocity $\mathbf{f}$ at position $\mathbf{r}$ and time t is represented by

$$\mathbf{f}(\mathbf{r}, t) = \sum_{m=1}^{M} \sum_{n=1}^{N} \eta_{(m,n)} \phi_m(\mathbf{r}) \psi_n(t), \tag{4}$$

where $\eta_{(m,n)} \in \mathbb{R}^2$ are constant parameter vectors. The total number of parameters, $\eta_{(m,n)}$, in this model is $M \times N$ for each of the x and y components of flow.

By assuming separation of variables (i.e., we assume that the flow field, as the solution to the geophysical PDEs, can be approximated by a space-dependent term multiplied by a time-dependent term), we can reduce the number of parameters associated with the flow model in Equation (4). Let us denote constant parameters coupled with the spatial basis functions for the x and y components of flow by $\eta_{x,m} \in \mathbb{R}$ and $\eta_{y,m} \in \mathbb{R}$, respectively. We also denote constant parameters coupled with the temporal basis functions for the x and y components of flow by $\rho_{x,n} \in \mathbb{R}$ and $\rho_{y,n} \in \mathbb{R}$, respectively. With M spatial basis functions and N temporal basis functions, the x and y components of flow velocity $\mathbf{f}$ at position $\mathbf{r}$ and time t is represented by

$$\begin{aligned} f_x(\mathbf{r}, t; \Theta_x) &= \left(\sum_{m=1}^{M} \eta_{x,m} \phi_m(\mathbf{r}) \right) \left(\sum_{n=1}^{N} \rho_{x,n} \psi_n(t) \right) \\ f_y(\mathbf{r}, t; \Theta_y) &= \left(\sum_{m=1}^{M} \eta_{y,m} \phi_m(\mathbf{r}) \right) \left(\sum_{n=1}^{N} \rho_{y,n} \psi_n(t) \right), \end{aligned} \tag{5}$$

where Θ_x and Θ_y denote all the parameters for the x and y components of flow, respectively (i.e., $\Theta_x = [\eta_x^T, \rho_x^T]^T$ and $\Theta_y = [\eta_y^T, \rho_y^T]^T$ for $\eta_x = [\cdots, \eta_{x,m}, \cdots]^T$, $\eta_y = [\cdots, \eta_{y,m}, \cdots]^T$, $\rho_x = [\cdots, \rho_{x,n}, \cdots]^T$, and $\rho_y = [\cdots, \rho_{y,n}, \cdots]^T$). Without loss of generality, we can express Equation (5) as

$$\begin{aligned} f_x(\mathbf{r}, t) &= h(\mathbf{r}, t; \Theta_x) \\ f_y(\mathbf{r}, t) &= h(\mathbf{r}, t; \Theta_y), \end{aligned} \tag{6}$$

where $h(\mathbf{r}, t; \cdot)$ is a nonlinear mapping from $\Theta_x \in \mathbb{R}^{M+N}$ to $f_x \in \mathbb{R}$ or from $\Theta_y \in \mathbb{R}^{M+N}$ to $f_y \in \mathbb{R}$. Compared to the flow model in Equation (4), the number of parameters decreases to $M + N$.

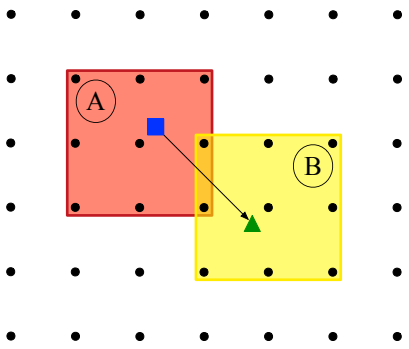


Figure 1: Illustration of the flow model initialization. The black round dots represent the grid points where data from flow data sources are available to initialize a data-driven flow model. The blue rectangle and the green triangle represent the previous and current positions of the center of vehicles, respectively.

Figure 1 illustrates an initialization process of the flow model that is used to guide a group of vehicles in a formation. The center of the initial vehicle positions is represented by the blue rectangle. To generate high-resolution flow data, we initialize our flow model using a historical time-series data set such as existing flow model output data with lower resolution. To initialize the model, we first select an area with $\kappa \times \kappa$ grid points (the black round dots) around vehicles from the historical data and refer to it as a *patch* (e.g., the red area, patch A). Then, we compute the spatial and temporal basis functions for the selected time and space of the historic data. The number of spatial basis functions, M, should be smaller than the number of grid points, κ^2, to avoid over-fitting. Given the computed basis functions and the historical data, we can find optimal parameters. Once initialized, the model can predict the flow around the vehicles to guide them and incorporate flow estimates from the vehicles to update parameters. However, when the vehicles move out of the selected patch (e.g., the green triangle in Figure 1), the spatial basis functions may have difficulty representing the spatial variation of flow outside the patch, so the model is reinitialized over a new patch (the yellow area, patch B).

2.2.2 Eulerian and Lagrangian Representations of Flow

In general, we describe the motion of flow using two approaches: Lagrangian and Eulerian. In the Lagrangian approach, we choose one particle and keep track of its position as it moves in space and time. The path traced out by this particle is called a particle pathline. An example is the ocean drifter which is an oceanographic device floating on the surface or at a particular depth of water to investigate ocean currents and water properties such as temperature or salinity. In contrast, the Eulerian approach is used to study the motion of particles moving in and out within fixed coordinates in space. This concept can be visualized by an image of a flow field such as ocean surface currents at a fixed time and is used as a general method for ocean circulation modeling. The lines comprising this flow field are called streamlines. While a pathline refers to the trace of a single particle in space and time, a streamline represents the line of motion of many particles at a fixed time.

These two representations of the flow field are strongly connected in the operation of UMSAs. To improve the sampling and navigation performance of UMSAs in the presence of flow, we incorporate predictions from ocean models (Eulerian representation of flow) into navigation through path planning, through which optimal trajectories under flow (Lagrangian representation of flow) are designed. In other words, path planning converts the Eulerian representation of the flow field to the Lagrangian representation.

The Lagrangian flow information is an important tool to understand the flow [35] and can be used to analyze the influence of ocean models on UMSAs [83]. For example, controlled Lagrangian particle tracking (CLPT) [83] examines the Lagrangian particle motion that is determined by ambient flow $\mathbf{f}$ and control input $\mathbf{u}$. Consider a particle with $\mathbf{r}(t)$ denoting its position at time t, whose motion is described by

$$d\mathbf{r} = (\mathbf{f}(\mathbf{r}, t) + \mathbf{u}(\mathbf{r}, t))dt. \tag{7}$$

Let $\mathbf{z}$ denote the position of a particle under modeled flow $\mathbf{F}_M$, which can be obtained by

$$d\mathbf{z} = (\mathbf{F}_M(\mathbf{z}, t) + \mathbf{u}(\mathbf{z}, t))dt. \tag{8}$$

Then, starting from $\mathbf{r}(0) = \mathbf{z}(0)$, the error $\mathbf{e}(t)$ along the trajectory is given by

$$\begin{aligned} d\mathbf{e} &= d\mathbf{r} - d\mathbf{z} \\ &= (\mathbf{f}(\mathbf{r}, t) + \mathbf{u}(\mathbf{r}, t) - \mathbf{F}_M(\mathbf{z}, t) - \mathbf{u}(\mathbf{z}, t))dt, \end{aligned} \tag{9}$$

which is referred to as the CLPT error. CLPT assumes perfect localization for UMSAs, providing a theoretical framework for understanding the interaction between flows predicted by ocean models and UMSAs.

In a similar way to CLPT, MT analyzes the difference between the actual vehicle trajectory under real flow and estimated vehicle trajectory under estimated flow. Unlike CLPT, MT focuses on practical analysis of the influence of flow on UMSAs with imperfect localization capabilities. Thus, MT assumes that actual vehicle position $\mathbf{r}$ is unknown and control input $\mathbf{u}$ in Equation (7) depends on estimated vehicle position $\mathbf{z}$ instead of actual vehicle position $\mathbf{r}$. Then, Equation (9) becomes $d\mathbf{e} = (\mathbf{f}(\mathbf{r}, t) - \mathbf{F}_M(\mathbf{z}, t))dt$, the integration of which gives the motion-integration error. MT estimates the Eulerian representation of flow from the motion-integration error along the vehicle trajectory. That is, MT converts the Lagrangian representation of the flow field to the Eulerian representation.

Figure 2 illustrates a flow diagram of maritime cyber data for autonomy in an operation of UMSAs associated with the development of an autonomous ocean sampling network [20, 28, 50]. Using ocean model predictions, the planning, control, and management of UMSAs are performed and the resulting navigation strategies and control inputs are provided to UMSAs. Then, combined with other observations from ocean observing systems such as satellites and moorings, data collected from UMSAs along their trajectories including vehicle trajectory information (Lagrangian flow data) are assimilated into ocean models (Eulerian flow data).

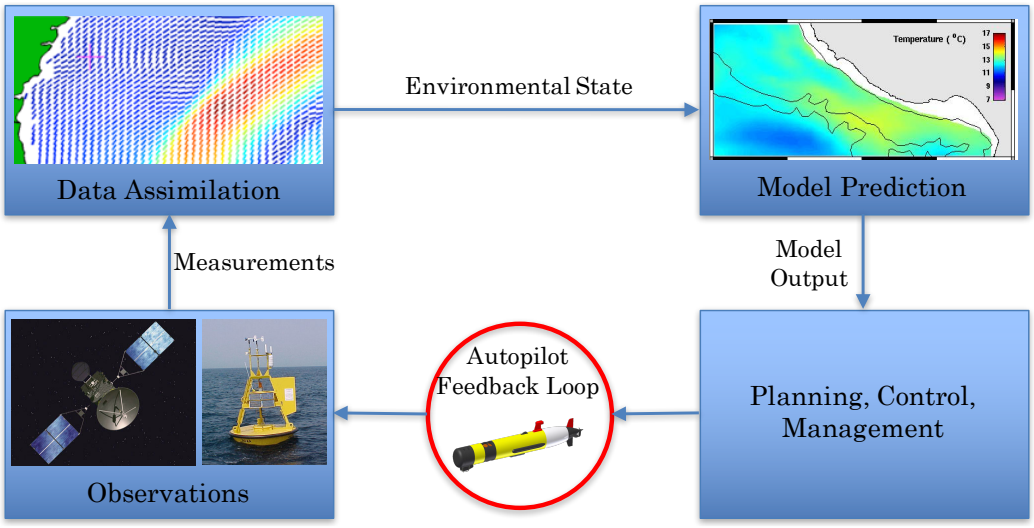


Figure 2: The maritime cyber cycle for autonomy in a typical operation of UMSAs.

2.2.3 Data Sources of Flow

In this document, we consider data sources of both Eulerian and Lagrangian flow. The following three of the typical data sources are reviewed: the high-frequency (HF) radar system, moorings, and gliders.

As stationary ocean observing technology, the HF-radar system [64, 31, 74] is widely used for the coastal sea surface current observation. The HF-radar system is a shore-based remote sensing system using the over-the-horizon radar technology. Shore stations of HF-radar emit radio signals that bounce off surface waves and return to the receiver. The received radio wave is used to compute ocean surface current movement relative to ocean surface wave movement. In general, hourly data with 3×3 or 6×6 km^2 spatial resolution are published online with a three hour processing delay. In addition to this operational limit, even though the HF-radar system provides Eulerian data with a large spatial coverage, its spatial resolution is considered low for operations of UMSAs.

A mooring is a stationary oceanographic instrumentation anchored on the sea floor and has a collection of sensors connected to a wire from the anchored station. A

mooring with the current meter can measure ocean currents typically every hour at different depths from near the sea surface to near the sea bottom. Although a mooring provides Eulerian data on fast time scales, since it is fixed at one location, its flow data provides insufficient spatial variability for UMSAs unless a set of moorings is installed in a very dense network or array.

In addition to the above Eulerian flow data sources, UMSAs are important tools for providing Lagrangian flow data. UMSAs typically have limited localization capabilities underwater [91], so their actual positions are only available when they are at the surface of water. Therefore, UMSAs in general follow predicted trajectories generated prior to diving into the water. Since the motion of a UMSA is perturbed by ambient flow, its actual trajectory deviates from its predicted trajectory, arising the *motion-integration error*. Some UMSAs estimate constant flow velocity along their previous trajectories based on the motion-integration error. Both this flow estimate and the motion-integration error itself are very important data that provide Lagrangian flow information.

2.3 Long Bay Field Experiments

Sustained wintertime phytoplankton blooms are observed off the coast of Long Bay, South Carolina in multi-year satellite chlorophyll images. In most areas of the South Atlantic Bight shelf, upwelling associated with Gulf Stream frontal eddies is a major mechanism of nutrient input, driving phytoplankton growth at the shelf edge. Just upstream of the Long Bay study area, the Charleston Bump deflects the Gulf Stream offshore, effectively shutting down direct frontal eddy nutrient input to Long Bay [49]. Prior *in-situ* observations [2] suggest recurring input of nutrients from the upper slope to the outer shelf off Long Bay from winter to early spring, but they do not provide evidence of a physical mechanism that accounts for both the alongshelf and seasonal patterns of the winter phytoplankton blooms associated with Long Bay.

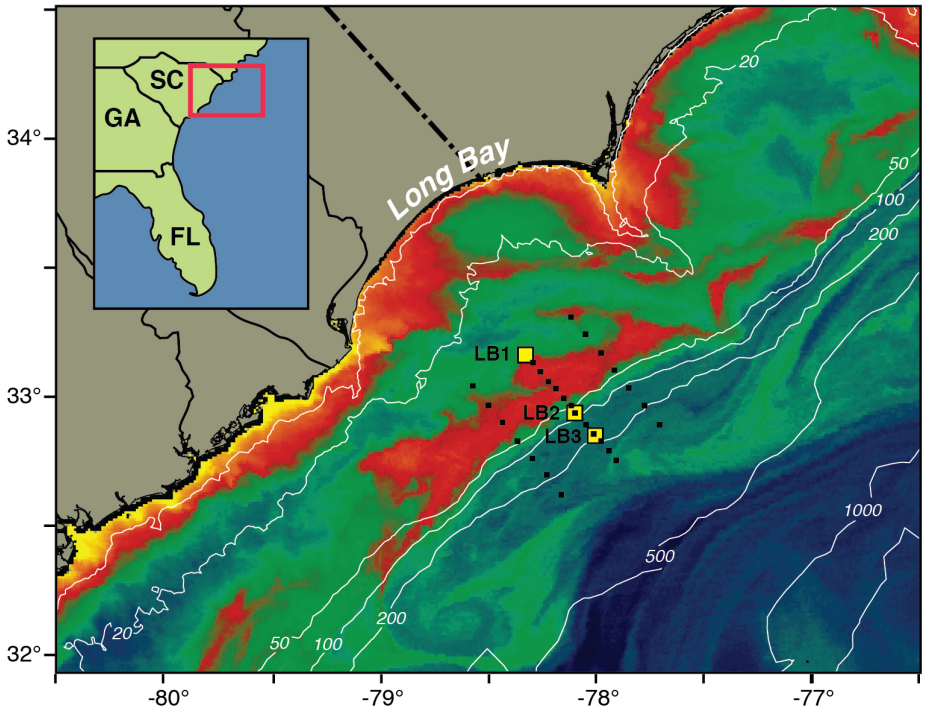


Figure 3: The survey domain near Long Bay, processed at Skidaway Institute of Oceanography based on MODIS (Moderate Resolution Imaging Spectroradiometer) Aqua chlorophyll imagery. High concentrations at the shelf break persist through much of winter and are distinct from high chlorophyll values adjacent to the coast. The figure shows the locations of moored instruments LB1, LB2, and LB3 with yellow rectangles and the three transect tracks of survey cruises with dotted lines in the cross-shore direction.

To study the mechanisms of phytoplankton bloom formation in the outer shelf to upper slope region near Long Bay (see Figure 3), we deployed various oceanic sensing platforms in a field experiment (LB2012) from January to April 2012. To observe the vertically-resolved property fluxes, we deployed three moored instruments at three locations on the shelf/slope LB1 (inner shelf, 30-m depth), LB2 (shelf break, 75-m depth), and LB3 (upper slope, 175-m depth). Near-continuous cross-shelf and upper slope observations were obtained by two Slocum gliders, which collected temperature, salinity, pressure, chlorophyll fluorescence, dissolved oxygen (DO), colored dissolved organic matter (CDOM), and turbidity data. One glider (*Ramses*) with maximum operable depth of 200 m operated as a virtual mooring, maintaining its position at

the edge of the Gulf Stream near LB3. The other glider (*Pelagia*) with maximum operable depth of 100 m conducted a cross-shelf section between LB1 and LB3 (or LB2 depending on ocean conditions). The combined data set from both gliders help us identify the physical and biological drivers of phytoplankton bloom formation in the study domain. In a follow-up experiment (LB2013) in late February 2013, one moored profiler sampled at LB2 and one Slocum glider (*Modena*) with maximum operable depth of 200 m between LB+20 (33-m depth) and LB+40 (130-m depth), which are 20 and 40 km offshore from LB1 in the cross-shore direction, respectively.

Underwater gliders are slowly moving vehicles whose through-water speed is typically 0.25–0.35 m/s; therefore, their motion is sensitive to ocean flow. In addition, since its underwater localization capability is limited, a glider navigates via dead reckoning and computes an estimate of averaged flow velocity along its trajectory between the last and current surfacing positions at the surface of water. This flow estimate is incorporated into dead reckoning from the current diving position until a glider reaches the next surfacing position. However, since ocean currents contain tidal components, the phase of this flow estimate will differ from that of real flow experienced by a glider underwater over the next surfacing interval, degrading navigation performance of a glider.

The observational domain, which covers the shelf and the shelf break off Long Bay, is characterized by strong tidal and Gulf Stream currents. Strong tides are largely aligned with the cross-shore direction over the shelf. Circulation at the shelf break is dominated by the Gulf Stream, which has highly variable and complex circulation because of eddies and filaments that develop and propagate along its shoreward front. The complex ocean dynamics in this region form steep temporal and spatial gradients in flow off the coast of Long Bay, and ocean currents here often exceed the forward speed of a glider. The typical speed of the tidal and Gulf Stream currents on the shelf is 0.2–0.3 and 0.5–0.75+ m/s, respectively. The maximum speed of glider-estimated

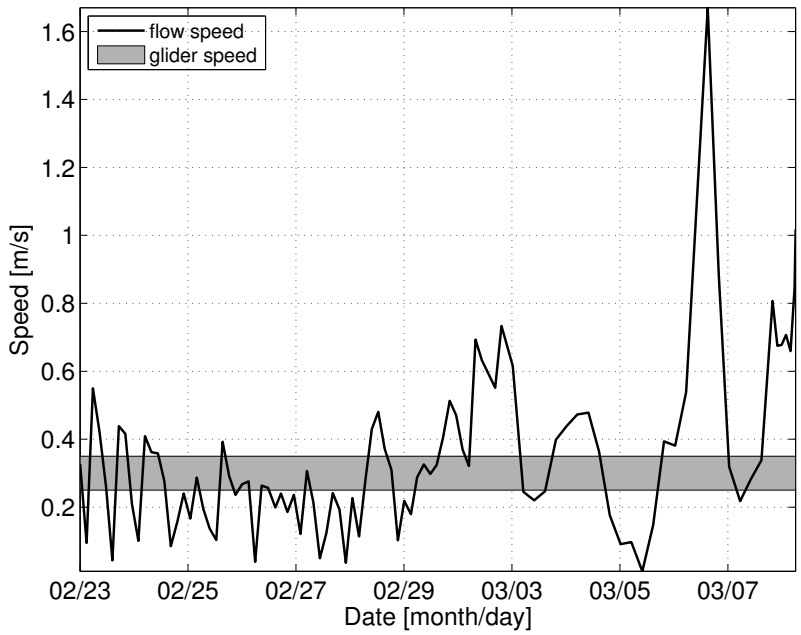


Figure 4: Flow speed near Long Bay. The black solid line is the speed of glider-estimated depth-averaged flow during the 2012 Long Bay deployment, and the gray shaded area indicates the range of the typical horizontal through-water speed of the glider.

flow, 1.67 m/s, in the Gulf Stream is shown in Figure 4. This challenging oceanic condition emphasized the importance of knowledge of flow in navigation of UMSAs.

To increase navigation accuracy of gliders in the presence of flow with steep temporal and spatial variability, we developed an empirical hybrid ocean model (EHOM) that combines a tidal ocean model with glider-estimated flow. Figure 5 displays the trajectories of two simulated gliders, one under the default glider navigation method with glider-estimated flow and the other under the proposed navigation method with EHOM, navigating a transect line (the red line) under a simulated flow field. After 14 surfacing events, both gliders managed to reach the target end point. Given waypoints generated using predicted flow via path planning, the glider driven by the proposed method (the dark blue line with rectangles) followed the transect line very closely under flow and reached the target end point smoothly. On the contrary, the

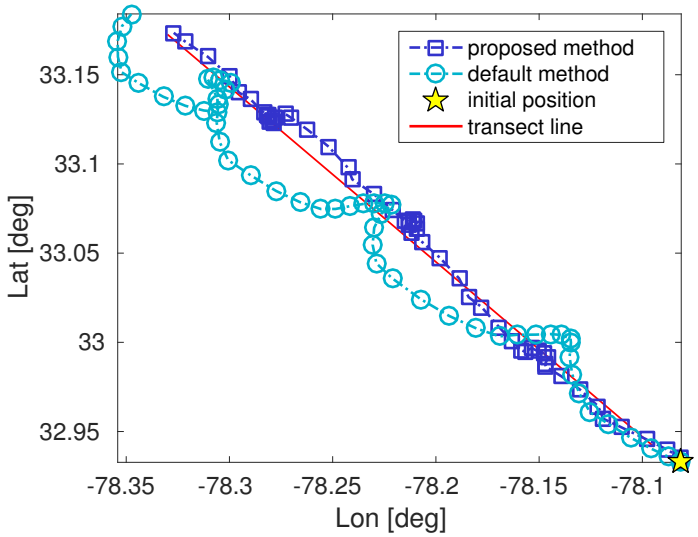


Figure 5: Trajectories of simulated gliders driven by the default and proposed navigation methods for the transect track (the red line). The dark blue line with rectangles is the trajectory of the glider using the proposed method with EHOM, and the light blue line with circles is the trajectory of the glider using the default navigation method with glider-estimated flow.

glider driven by the default navigation method (the light blue line with circles) meandered along the transect line. This result shows that the proposed method can provide more precise control under flow. In addition to the simulation results, the method successfully guided multiple underwater gliders during Long Bay field experiments. This work inspired a further study for flow field modeling and estimation for guidance of UMSAs.

CHAPTER III

MOTION TOMOGRAPHY

3.1 Formulation of MT

The MT method [88, 9, 8] estimates a flow field based on the motion-integration error and trajectory information of UMSAs. In this section, for easy analysis and clarity, we simplify the problem by making the following assumptions:

Assumption 1. *The flow field is time-invariant over one observation interval $\mathcal{T}$. As a result, the flow is now represented by $\mathbf{f}(\mathbf{r})$ instead of $\mathbf{f}(\mathbf{r}, t)$.*

Assumption 2. *The horizontal through-water speeds (as opposed to ground speeds) of all vehicles are identically s_{h}, which is a constant.*

These assumptions are often realistic for oceanic applications under certain conditions. Assumption 1 can be satisfied if we choose $\mathcal{T}$ such that the error caused by a time-varying flow field is minimal. For Assumption 2, let us consider that the observation interval of UMSAs is mostly several hours, leading to travel distances of a few km. Considering the travel distances of UMSAs, the effect of variations of the vehicle speed is often trivial. In Section 3.5, we provide preliminary analysis on the violations of these assumptions through experiments with mobile platforms in synthetic and real flow fields.

Suppose we deploy K vehicles with $\mathbf{r}_i$, $i = \{1, \cdots, K\}$ denoting their positions in domain $\mathcal{D}$ (see Figure 6). After observation interval $\mathcal{T}$, we obtain the motion-integration errors $\mathbf{d}_i$ of the vehicles. We denote the trajectories of each vehicle by curve γ_i according to (3) and introduce arc-length parameter ℓ_i for curve γ_i, given by

$$d\ell_i = s_{\mathrm{tr}}(\mathbf{f}(\mathbf{r}_i))dt, \tag{10}$$

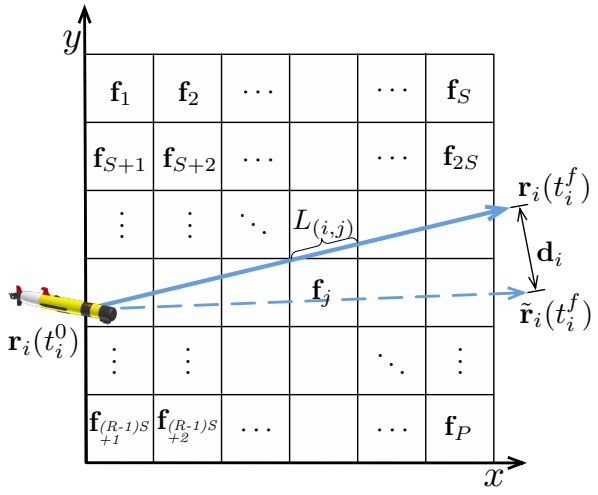


Figure 6: Illustration of MT mapping formulation. Actual (the blue solid line) and predicted (the blue dashed line) vehicle trajectories are displayed in a discretized domain.

in which s_{tr} is the speed of the vehicle along its actual trajectory, which satisfies

$$s_{\text{tr}}(\mathbf{f}(\mathbf{r}_i)) = \|\dot{\mathbf{r}}_i\| = \left\| s_{\text{h}} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} + \mathbf{f}(\mathbf{r}_i) \right\|. \tag{11}$$

Substituting Equation (10) into Equation (3), we derive

$$\mathbf{d}_i = \int_{\gamma_i} \frac{1}{s_{\text{tr}}(\mathbf{f}(\mathbf{r}_i))} \mathbf{f}(\mathbf{r}_i) d\ell_i. \tag{12}$$

If $s_{\text{tr}} = 0$, Equation (12) is not well-defined. The equality $s_{\text{tr}} = 0$ simply means that the vehicle motion cancels out the flow, leading to a vehicle staying at the previous position. Without loss of generality, we can assume that $s_{\text{tr}} \neq 0$ by a controller design, or since $s_{\text{tr}} = 0$ does not affect the continuity and smoothness of the vehicle trajectory, we can evaluate the integral in Equation (12) over the trajectory $\gamma_i = \{\mathbf{r}_i(t) : s_{\text{tr}}(\mathbf{r}_i) \neq 0\}_{t \in \mathcal{T}}$.

Let us discretize the domain $\mathcal{D}$ into $P = R \times S$ grid cells with $D_{(r,s)}$ denoting the (r,s)th cell. We define index $j = (r-1)S + s$ such that $D_j \equiv D_{(r,s)}$, $j = \{1, \cdots, P\}$. We denote the flow velocity in D_j by $\mathbf{f}_j$ and assume $\mathbf{f}_j$ is constant. For the ith vehicle

in D_j, we assume that vehicle heading $\theta_{(i,j)}$ is constant within D_j. Then, the speed of the ith vehicle along the trajectory is given by

$$s_{\text{tr}}^{(i,j)}(\mathbf{f}_j) = \left\| s_{\text{h}} \begin{bmatrix} \cos\theta_{(i,j)} \\ \sin\theta_{(i,j)} \end{bmatrix} + \mathbf{f}_j \right\|, \tag{13}$$

which leads to the linear trajectory within D_j and the piecewise linear trajectory over the domain $\mathcal{D}$. We assume that vehicle heading $\theta_{(i,j)}$ is available to us or can be estimated with small bounded error. The impact of vehicle heading with small bounded error on the results on MT is discussed in Section 3.2.

For the ith vehicle passing through the jth cell, D_j, the length of the vehicle trajectory, $L_{(i,j)}$, can be obtained by

$$L_{(i,j)} = \int_{\gamma_i[D_j]} d\ell_i, \tag{14}$$

in which $\gamma_i[D_j]$ represents curve γ_i within the spatial interval for D_j in a planar space. Let us denote the time that the ith vehicle enters D_j by $t^0_{(i,j)}$ and the time that the ith vehicle leaves D_j by $t^f_{(i,j)}$. Let $\mathbf{r}^0_{(i,j)} = \mathbf{r}_i\big(t^0_{(i,j)}\big)$ and $\mathbf{r}^f_{(i,j)} = \mathbf{r}_i\big(t^f_{(i,j)}\big)$. Since the trajectory within a cell is linear, Equation (14) is equivalent to

$$L_{(i,j)} = \left\| \mathbf{r}^f_{(i,j)} - \mathbf{r}^0_{(i,j)} \right\|. \tag{15}$$

The position $\mathbf{r}^f_{(i,j)}$ can be obtained by

$$\mathbf{r}^f_{(i,j)} = \mathbf{r}^0_{(i,j)} + \int_{t^0_{(i,j)}}^{t^f_{(i,j)}} \left(s_{\text{h}} \begin{bmatrix} \cos\theta_{(i,j)} \\ \sin\theta_{(i,j)} \end{bmatrix} + \mathbf{f}_j \right) dt. \tag{16}$$

From Equation (16), we see that $\mathbf{r}^f_{(i,j)}$ depends on flow $\mathbf{f}_j$. Suppose prior to D_j, the ith vehicle passed D_{j-1}, i.e., $t^0_{i,j} = t^f_{i,j-1}$. Then,

$$\begin{aligned} \mathbf{r}^0_{(i,j)} &= \mathbf{r}^f_{(i,j-1)} \\ &= \mathbf{r}^0_{(i,j-1)} + \int_{t^0_{(i,j-1)}}^{t^f_{(i,j-1)}} \left(s_{\text{h}} \begin{bmatrix} \cos\theta_{(i,j-1)} \\ \sin\theta_{(i,j-1)} \end{bmatrix} + \mathbf{f}_{j-1} \right) dt, \end{aligned} \tag{17}$$

which shows that $\mathbf{r}^0_{(i,j)}$ is affected by flow $\mathbf{f}_{j-1}$. Following the same procedure for the cells along the past vehicle trajectory, we find that $\mathbf{r}^0_{(i,j)}$ depends on all the flow along the past trajectory. Therefore, to compute $L_{(i,j)}$, we should recursively consider the influence of flow along the vehicle trajectory from $t = t_i^0$ to $t^f_{(i,j)}$. Let us stack all the flow velocities for the grid cells and define vector $\overline{\mathbf{f}} = [\overline{\mathbf{f}}_x^T, \overline{\mathbf{f}}_y^T]^T$, where $\overline{\mathbf{f}}_x = [f_{x,1}, f_{x,2}, \cdots, f_{x,P}]^T$ is the x component of all the flow velocities for the grid cells and $\overline{\mathbf{f}}_y = [f_{y,1}, f_{y,2}, \cdots, f_{y,P}]^T$ is the y component of all the flow velocities for the grid cells. To consider the dependency of the vehicle trajectory on the flow field, we express the length of the vehicle trajectory as $L_{(i,j)} = L_{(i,j)}(\overline{\mathbf{f}})$.

Based on the discretization setting designed above, Equation (12) can be discretized into

$$\mathbf{d}_i = \sum_{j=1}^{P} \frac{L_{(i,j)}(\overline{\mathbf{f}})}{s_{\text{tr}}^{(i,j)}(\mathbf{f}_j)} \mathbf{f}_j, \quad i = \{1, \cdots, K\}. \tag{18}$$

Considering the flow velocity along the x and y directions separately, we have

$$\begin{aligned} d_{x,i} &= \sum_{j=1}^{P} \frac{L_{(i,j)}(\overline{\mathbf{f}})}{s_{\text{tr}}^{(i,j)}(\mathbf{f}_j)} f_{x,j}, \\ d_{y,i} &= \sum_{j=1}^{P} \frac{L_{(i,j)}(\overline{\mathbf{f}})}{s_{\text{tr}}^{(i,j)}(\mathbf{f}_j)} f_{y,j}. \end{aligned} \tag{19}$$

By constructing vectors $\mathbf{d}_x = [d_{x,1}, d_{x,2}, \cdots, d_{x,K}]^T$ and $\mathbf{d}_y = [d_{y,1}, d_{y,2}, \cdots, d_{y,K}]^T$, we can rewrite Equation (19) as

$$\begin{aligned} \mathbf{d}_x &= \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}}_x \\ \mathbf{d}_y &= \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}}_y, \end{aligned} \tag{20}$$

where $\overline{\mathbf{f}} = [\overline{\mathbf{f}}_x^T, \overline{\mathbf{f}}_y^T]^T$ and

$$\mathbf{L}(\overline{\mathbf{f}}) = \begin{bmatrix} \frac{L_{(1,1)}(\overline{\mathbf{f}})}{s_{\text{tr}}^{(1,1)}(\mathbf{f}_1)} & \cdots & \frac{L_{(1,P)}(\overline{\mathbf{f}})}{s_{\text{tr}}^{(1,P)}(\mathbf{f}_P)} \\ \vdots & \ddots & \vdots \\ \frac{L_{(K,1)}(\overline{\mathbf{f}})}{s_{\text{tr}}^{(K,1)}(\mathbf{f}_1)} & \cdots & \frac{L_{(K,P)}(\overline{\mathbf{f}})}{s_{\text{tr}}^{(K,P)}(\mathbf{f}_P)} \end{bmatrix}. \tag{21}$$

$\mathbf{L}(\overline{\mathbf{f}})$ contains information of how long individual vehicles stay in each grid cell given

flow $\overline{\mathbf{f}}$ and Equation (20) is nonlinear and typically underdetermined (i.e., the number of equations, K, is less than the number of unknowns, P).

Remark 1. *MT solves an inverse problem for* $\mathbf{f}$ *from Equation* (20)*. Even though the equation follows a similar form to the classical CT problem, the nonlinearity is a significant difference between CT and MT, which brings challenges to MT. If* $\tilde{\mathbf{f}} \neq 0$ *in obtaining Equation* (3)*, the motion-integration error is caused by the difference between real flow* $\mathbf{f}$ *and estimated (or predicted) flow* $\tilde{\mathbf{f}}$*. In this case,* $\overline{\mathbf{f}}_x$ *and* $\overline{\mathbf{f}}_y$ *are constructed such that* $\overline{\mathbf{f}}_x = [f_{x,1} - \tilde{f}_x, f_{x,2} - \tilde{f}_x, \cdots, f_{x,P} - \tilde{f}_x]^T$ *and* $\overline{\mathbf{f}}_y = [f_{y,1} - \tilde{f}_y, f_{y,2} - \tilde{f}_y, \cdots, f_{y,P} - \tilde{f}_y]^T$*, for which MT solves the equations in Equation* (20)*.*

To solve the equations in Equation (20) for flow $\overline{\mathbf{f}}$, we first need to determine $\mathbf{L}(\overline{\mathbf{f}})$ in Equation (21) based on knowledge of vehicle trajectories. However, because of limited localization capabilities of UMSAs, their real trajectories under flow are often unknown and thus must be estimated before solving Equation (20). We refer to this key step as trajectory tracing. The idea of trajectory tracing is to estimate vehicle trajectories $\tilde{\gamma}$ under flow using the best estimate of the flow field. Therefore, flow field mapping through MT is an iterative process consisting of two key steps: trajectory tracing and flow field estimation.

3.2 Trajectory Tracing and Its Error Bound

To address the problem of unknown nonlinear trajectories, we trace vehicle trajectories by iteratively simulating the vehicle trajectory, $\tilde{\gamma}$, while updating the current estimate of the flow field, $\tilde{\mathbf{f}}$. For the ith vehicle, its traced trajectory $\tilde{\gamma}_i$ is obtained by using

$$\dot{\tilde{\mathbf{r}}}_i(t) = s_{\mathrm{h}} \begin{bmatrix} \cos\tilde{\theta} \\ \sin\tilde{\theta} \end{bmatrix} + \tilde{\mathbf{f}}(\tilde{\mathbf{r}}_i), \tag{22}$$

in which $\tilde{\theta} \in \{\theta_{(i,j)}\}$ and $\tilde{\mathbf{f}} \in \{\mathbf{f}_j\}$, $i = \{1, \cdots, K\}$, $j = \{1, \cdots, P\}$ are piecewise constant vehicle heading estimate and piecewise constant flow estimate for the discretized domain $\mathcal{D} = \cup_j D_j$, respectively. To increase the accuracy of the traced vehicle trajectory, we iteratively alternate between flow estimation and trajectory tracing.

Suppose a vehicle navigates over observation interval $\mathcal{T} = [t^0, t^f]$ with observation time horizon $T_{\text{obs}} = t^f - t^0 > 0$. Let γ be the true vehicle trajectory which is obtained by

$$\gamma = \int_{t^0}^{t^f} \left(s_{\text{h}} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + \mathbf{f}(\mathbf{r}) \right) dt, \tag{23}$$

where s_{h} is the horizontal vehicle speed, $\theta \in [-\pi, \pi]$ is the actual vehicle heading, and $\mathbf{f}$ is the true flow field. Let $\tilde{\gamma}$ be the traced vehicle trajectory which is obtained by

$$\tilde{\gamma} = \int_{t^0}^{t^f} \left(s_{\text{h}} \begin{bmatrix} \cos\tilde{\theta} \\ \sin\tilde{\theta} \end{bmatrix} + \tilde{\mathbf{f}}(\mathbf{r}) \right) dt, \tag{24}$$

where $\tilde{\theta} \in [-\pi, \pi]$ is the estimated vehicle heading and $\tilde{\mathbf{f}}$ is the estimated flow field. The actual heading and the estimated heading may differ because of the error that may come from measurement error of a compass or piecewise linearization error of the heading in the discretization setting. We make the following assumption on $\tilde{\theta} - \theta$.

Assumption 3. *For all t, $|\tilde{\theta} - \theta|$ is bounded above by a small constant $0 \leq \epsilon \ll \pi$.*

Lemma 1. *If $|\tilde{\theta} - \theta| \leq \epsilon$, then $1 - \frac{1}{2}(\tilde{\theta} - \theta)^2 \leq \cos(\tilde{\theta} - \theta)$.*

Proof. Let us define $\phi = \tilde{\theta} - \theta$. Since all functions of ϕ on the both sides of the inequality are even about 0, let us consider only the case in which $0 \leq \phi \leq \pi$. By Taylor's theorem, $f(\phi) = cos(\phi) = \mathcal{P}_n + \mathcal{R}_n$, in which $\mathcal{P}_n$ is the nth order Taylor polynomial for $cos(\phi)$ at $\phi = c$ and $\mathcal{R}_n = \int_c^{\phi} \frac{f^{(n+1)}(\alpha)}{n!}(\phi - \alpha)^n d\alpha$ is the remainder. For $n = 2$ and $c = 0$, we have $cos(\phi) = 1 - \frac{1}{2}\phi^2 + \frac{1}{2}\int_0^{\phi} \sin(\alpha)(\phi - \alpha)^2 d\alpha$. Since $\frac{1}{2}\int_0^{\phi} \sin(\alpha)(\phi - \alpha)^2 d\alpha \geq 0$, the inequality holds. □

In Theorem 1 below, we prove that given that $\|\tilde{\mathbf{f}}(\mathbf{r}) - \mathbf{f}(\mathbf{r})\|$ is uniformly bounded above, the difference between real and traced vehicle trajectories is bounded. For a discretized domain $\mathcal{D}$, let us suppose that the estimated flow map by MT is denoted by $\tilde{\overline{\mathbf{f}}}$ and the true flow map is denoted by $\overline{\mathbf{f}}^{\star}$. In later sections, we prove that $\|\tilde{\overline{\mathbf{f}}} - \overline{\mathbf{f}}^{\star}\|$ converges to zero in the L2 sense by using the flow field estimation algorithms developed for both non-parametric and parametric MT.

Theorem 1. *Given that Assumption 3 holds and* $\|\tilde{\mathbf{f}} - \mathbf{f}\| \leq \delta$, $\|\tilde{\gamma} - \gamma\|$ *is bounded above by* $(\epsilon s_{\mathrm{h}} + \delta)T_{\mathrm{obs}}$, *in which* s_{h} *is the horizontal vehicle speed and* T_{obs} *is the observation time horizon.*

Proof. Let us define error term $e = \tilde{\gamma} - \gamma$. Then, we have

$$
\begin{aligned}
e &= \tilde{\gamma} - \gamma \\
&= \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \begin{bmatrix} \cos\tilde{\theta} \\ \sin\tilde{\theta} \end{bmatrix} + \tilde{\mathbf{f}}(\mathbf{r}) \right) dt - \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + \mathbf{f}(\mathbf{r}) \right) dt \\
&= \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \begin{bmatrix} \cos\tilde{\theta} - \cos\theta \\ \sin\tilde{\theta} - \sin\theta \end{bmatrix} + \tilde{\mathbf{f}}(\mathbf{r}) - \mathbf{f}(\mathbf{r}) \right) dt.
\end{aligned}
$$

The Euclidean norm of the error is

$$
\begin{aligned}
\|e\| &= \left\| \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \begin{bmatrix} \cos\tilde{\theta} - \cos\theta \\ \sin\tilde{\theta} - \sin\theta \end{bmatrix} + \tilde{\mathbf{f}}(\mathbf{r}) - \mathbf{f}(\mathbf{r}) \right) dt \right\| \\
&\leq \int_{t^0}^{t^f} \left(\left\| s_{\mathrm{h}} \begin{bmatrix} \cos\tilde{\theta} - \cos\theta \\ \sin\tilde{\theta} - \sin\theta \end{bmatrix} + \tilde{\mathbf{f}}(\mathbf{r}) - \mathbf{f}(\mathbf{r}) \right\| \right) dt \\
&\leq \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \left\| \begin{bmatrix} \cos\tilde{\theta} - \cos\theta \\ \sin\tilde{\theta} - \sin\theta \end{bmatrix} \right\| + \left\| \tilde{\mathbf{f}}(\mathbf{r}) - \mathbf{f}(\mathbf{r}) \right\| \right) dt \\
&\leq \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \left\| \begin{bmatrix} \cos\tilde{\theta} - \cos\theta \\ \sin\tilde{\theta} - \sin\theta \end{bmatrix} \right\| + \delta \right) dt
\end{aligned}
$$

$$
\begin{aligned}
&= \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \sqrt{(\cos\tilde{\theta} - \cos\theta)^2 + (\sin\tilde{\theta} - \sin\theta)^2} + \delta \right) dt \\
&= \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \sqrt{\cos^2\tilde{\theta} - 2\cos\tilde{\theta}\cos\theta + \cos^2\theta + \sin^2\tilde{\theta} - 2\sin\tilde{\theta}\sin\theta + \sin^2\theta} + \delta \right) dt \\
&= \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \sqrt{2 - 2\cos\tilde{\theta}\cos\theta - 2\sin\tilde{\theta}\sin\theta} + \delta \right) dt \\
&= \int_{t^0}^{t^f} \left(s_{\mathrm{h}} \sqrt{2 - 2\cos(\tilde{\theta} - \theta)} + \delta \right) dt.
\end{aligned}
$$

By applying Lemma 1, we have

$$
\begin{aligned}
\|e\| &\leq \int_{t^0}^{t^f} s_{\mathrm{h}} \sqrt{2 - 2\left(1 - \frac{1}{2}(\tilde{\theta} - \theta)^2\right)} dt + \int_{t^0}^{t^f} \delta dt \\
&= s_{\mathrm{h}} \int_{t^0}^{t^f} |\tilde{\theta} - \theta| dt + \int_{t^0}^{t^f} \delta dt \\
&\leq s_{\mathrm{h}} \int_{t^0}^{t^f} \epsilon dt + \int_{t^0}^{t^f} \delta dt \\
&= (\epsilon s_{\mathrm{h}} + \delta)(t^f - t^0) \\
&= (\epsilon s_{\mathrm{h}} + \delta) T_{\mathrm{obs}}.
\end{aligned}
$$

Therefore, given $\|\tilde{\mathbf{f}} - \mathbf{f}\| \leq \delta$, $\|\tilde{\gamma} - \gamma\| \leq (\epsilon s_{\mathrm{h}} + \delta) T_{\mathrm{obs}}$. □

3.3 A Kaczmarz-Type Method for Flow Field Estimation by MT

Because of the dependency of the vehicle trajectory on flow and the nonavailability of the unknown underwater vehicle trajectory in constructing $\mathbf{L}(\bar{\mathbf{f}})$, solving Equation (20) using gradient-based methods is complicated. Since MT shares a similar framework to CT, we derive an iterative flow field estimation algorithm referred to as a Kaczmarz-type method based on the Kaczmarz method [41, 42], an iterative method for solving a linear system of equations. The Kaczmarz method, also known as the algebraic reconstruction technique [30] in the medical imaging community, has been used for computerized tomography [43, 61, 36, 16].

Suppose $\bar{\mathbf{f}}^{\star}$ is a true solution to a linear system of equations, $\mathbf{L}\bar{\mathbf{f}} = \mathbf{d}$, where $\mathbf{L} \in \mathbb{R}^{K \times P}$, $\bar{\mathbf{f}} \in \mathbb{R}^{P}$, and $\mathbf{d} \in \mathbb{R}^{K}$. Let us denote the index for the iteration by k. As a

row-action method [6], given the kth approximation, $\overline{\mathbf{f}}^k$, of $\overline{\mathbf{f}}^\star$, the Kaczmarz method iterates the following optimization process:

$$\begin{aligned} \overline{\mathbf{f}}^{k+1} &= \underset{\overline{\mathbf{f}}}{\operatorname{argmin}} \frac{1}{2} \left\| \overline{\mathbf{f}} - \overline{\mathbf{f}}^k \right\|^2 \\ &\text{subject to} \quad d_i = \mathbf{L}_i \overline{\mathbf{f}}, \end{aligned} \tag{25}$$

where $\mathbf{L}_i$ is the ith row of matrix $\mathbf{L}$, d_i the ith element of vector $\mathbf{d}$, and $i = \text{mod}\ (k, K) + 1$, $k = \{0, 1, 2, \cdots\}$.

This optimization problem can be solved by the Lagrange multiplier method such that

$$\overline{\mathbf{f}}^{k+1} = \underset{\overline{\mathbf{f}}}{\operatorname{argmin}} \left[\frac{1}{2} \left(\left\| \overline{\mathbf{f}} - \overline{\mathbf{f}}^k \right\|^2 \right) + \lambda \left(d_i - \mathbf{L}_i \overline{\mathbf{f}} \right) \right],$$

where λ is the Lagrange multiplier. Let $\mathcal{L}(\overline{\mathbf{f}}, \lambda) = \frac{1}{2} \left(\left\| \overline{\mathbf{f}} - \overline{\mathbf{f}}^k \right\|^2 \right) + \lambda \left(d_i - \mathbf{L}_i \overline{\mathbf{f}} \right)$ be Lagrangian. The first order necessary condition for optimality is $\frac{\partial \mathcal{L}(\overline{\mathbf{f}}, \lambda)}{\partial \overline{\mathbf{f}}} = 0$. By taking the derivative of the Lagrangian with respect to $\overline{\mathbf{f}}$, we obtain

$$\frac{\partial \mathcal{L}(\overline{\mathbf{f}}, \lambda)}{\partial \overline{\mathbf{f}}} = \overline{\mathbf{f}} - \overline{\mathbf{f}}^k - \mathbf{L}_i^T \lambda = 0, \tag{26}$$

which leads to

$$\mathbf{L}_i^T \lambda = \overline{\mathbf{f}} - \overline{\mathbf{f}}^k. \tag{27}$$

By multiplying $\mathbf{L}_i$ on the both sides of the equation, it becomes

$$\mathbf{L}_i \mathbf{L}_i^T \lambda = d_i - \mathbf{L}_i \overline{\mathbf{f}}^k, \tag{28}$$

which gives

$$\lambda = \frac{d_i - \mathbf{L}_i \overline{\mathbf{f}}^k}{\|\mathbf{L}_i\|^2}. \tag{29}$$

By substituting (29) into (27) and letting $\overline{\mathbf{f}}$ be $\overline{\mathbf{f}}^{k+1}$, we obtain the following key formula:

$$\overline{\mathbf{f}}^{k+1} = \overline{\mathbf{f}}^k + \lambda_k \frac{d_i - \mathbf{L}_i \overline{\mathbf{f}}^k}{\|\mathbf{L}_i\|^2} \mathbf{L}_i^T, \tag{30}$$

where λ_k is a relaxation parameter that affects the convergence rate of the method.

Typically, the Kaczmarz method processes every K iterations from $i = 1$ to K as a batch. The Kaczmarz method is geometrically represented as an iterative projection of $\bar{\mathbf{f}}^k$ onto hyperplanes $\mathcal{H}_i = \{\bar{\mathbf{f}} \in \mathbb{R}^n | \mathbf{L}_i \bar{\mathbf{f}} = d_i\}$. Its convergence to a least squares solution to a consistent system is proved in [84, 70].

Extended from the Kaczmarz method, a method for MT has been developed in the author's previous work [88, 9, 8] to solve (20). For simplicity of presentation, let us omit x and y in the system, i.e., we have

$$\mathbf{d} = \mathbf{L}(\bar{\mathbf{f}})\bar{\mathbf{f}}, \tag{31}$$

where

$$\mathbf{L}(\bar{\mathbf{f}}) = \begin{bmatrix} \mathbf{L}_1(\bar{\mathbf{f}}) \\ \vdots \\ \mathbf{L}_K(\bar{\mathbf{f}}) \end{bmatrix}, \ \bar{\mathbf{f}} = \begin{bmatrix} f_1 \\ \vdots \\ f_P \end{bmatrix}, \ \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix}.$$

The method solves the following iterative sub-optimization problem:

$$\begin{aligned} \bar{\mathbf{f}}^{k+1} &= \underset{\bar{\mathbf{f}}}{\operatorname{argmin}} \frac{1}{2} \left\| \bar{\mathbf{f}} - \bar{\mathbf{f}}^k \right\|^2 \\ \text{subject to} \quad & d_i = \mathbf{L}_i(\bar{\mathbf{f}}^k)\bar{\mathbf{f}}, \end{aligned} \tag{32}$$

where $\mathbf{L}_i$ is the ith row of $\mathbf{L}$, d_i is the ith element of $\mathbf{d}$, $i = \mathrm{mod}(k, K) + 1$, and $k = \{0, 1, 2, \cdots\}$ is the iteration index. Note that to deal with the nonlinearity in (31), the constraint is linearized by using the latest estimate of $\bar{\mathbf{f}}$ at each iteration.

As is the case with the Kaczmarz method, this optimization problem can be solved by the Lagrange multiplier method such that

$$\bar{\mathbf{f}}^{k+1} = \underset{\bar{\mathbf{f}}}{\operatorname{argmin}} \left[\frac{1}{2} \left(\left\| \bar{\mathbf{f}} - \bar{\mathbf{f}}^k \right\|^2 \right) + \lambda \left(d_i - \mathbf{L}_i(\bar{\mathbf{f}}^k)\bar{\mathbf{f}} \right) \right],$$

where λ is the Lagrange multiplier. Let $\mathcal{L}(\bar{\mathbf{f}}, \lambda) = \frac{1}{2} \left(\left\| \bar{\mathbf{f}} - \bar{\mathbf{f}}^k \right\|^2 \right) + \lambda \left(d_i - \mathbf{L}_i(\bar{\mathbf{f}}^k)\bar{\mathbf{f}} \right)$ be Lagrangian. The first order necessary condition for optimality is $\frac{\partial \mathcal{L}(\bar{\mathbf{f}}, \lambda)}{\partial \bar{\mathbf{f}}} = 0$. By taking the derivative of the Lagrangian with respect to $\bar{\mathbf{f}}$, we obtain

$$\frac{\partial \mathcal{L}(\bar{\mathbf{f}}, \lambda)}{\partial \bar{\mathbf{f}}} = \bar{\mathbf{f}} - \bar{\mathbf{f}}^k - \left(\mathbf{L}_i(\bar{\mathbf{f}}^k) \right)^T \lambda = 0, \tag{33}$$

which leads to

$$\left(\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)^T \lambda = \overline{\mathbf{f}} - \overline{\mathbf{f}}^k. \tag{34}$$

By multiplying $\mathbf{L}_i(\overline{\mathbf{f}}^k)$ on the both sides of the equation, it becomes

$$\mathbf{L}_i(\overline{\mathbf{f}}^k)\left(\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)^T \lambda = d_i - \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k, \tag{35}$$

which gives

$$\lambda = \frac{d_i - \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2}. \tag{36}$$

By substituting (36) into (34) and letting $\overline{\mathbf{f}}$ be $\overline{\mathbf{f}}^{k+1}$, we obtain the following key formula:

$$\overline{\mathbf{f}}^{k+1} = \overline{\mathbf{f}}^k + \frac{d_i - \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2}\left(\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)^T. \tag{37}$$

As a Kaczmarz-type method, the method based on Equation (37) can be geometrically represented as an iterative projection of $\overline{\mathbf{f}}^k$ onto hyperplanes $\mathcal{H}_i = \{\overline{\mathbf{f}} \in \mathbb{R}^n | \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}} = d_i\}$.

Remark 2. *To solve* (32)*, one may consider an optimization problem*

$$\begin{aligned} &\textit{minimize} \quad \frac{1}{2}\|\overline{\mathbf{f}}\|^2 \\ &\textit{subject to} \quad \mathbf{d} - \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}} = 0, \end{aligned} \tag{38}$$

which can be viewed as a nonlinear least squares problem. However, the nonlinearity in (31) *may cause expensive gradient computation or lead to non-convexity of the problem. Therefore, we employ an iterative row-action approach and linearize the constraint at each iteration, yielding a convex optimization problem.*

Based on the Kaczmarz-type method for MT in Equation (37), we obtain a flow field estimation algorithm (Algorithm 1). In the algorithm, the updating equation includes a relaxation parameter, λ, which affects the convergence rate. We define residuals $\mathbf{r}^k_{\overline{\mathbf{f}},x} = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k_x - \mathbf{d}_x$ and $\mathbf{r}^k_{\overline{\mathbf{f}},y} = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k_y - \mathbf{d}_y$ for k that satisfies $\mathrm{mod}\,(k, K) + 1 = K$ to check for the convergence. We continue updating the solutions until the

Algorithm 1: MT flow field estimation

Data: Motion-integration errors $\mathbf{d}_i, i = \{1, \cdots, K\}$

Set $k = 0$. Make an initial guess of the solutions, $\overline{\mathbf{f}}_x^0$ and $\overline{\mathbf{f}}_y^0$.

repeat

  for $i = 1$ **to** K **do**

    Update the solutions by

$$\overline{\mathbf{f}}_x^{k+1} = \overline{\mathbf{f}}_x^k + \lambda^k \frac{d_{x,i} - \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_x^k}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2} \left(\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)^T,$$

$$\overline{\mathbf{f}}_y^{k+1} = \overline{\mathbf{f}}_y^k + \lambda^k \frac{d_{y,i} - \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_y^k}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2} \left(\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)^T.$$

    Let $k = k + 1$.

  end

  $\mathbf{r}_{\overline{\mathbf{f}},x}^k = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_x^k - \mathbf{d}_x$.

  $\mathbf{r}_{\overline{\mathbf{f}},y}^k = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_y^k - \mathbf{d}_y$.

until *a stopping condition is met (e.g.,* $\|\mathbf{r}_{\overline{\mathbf{f}},x}^k\|, \|\mathbf{r}_{\overline{\mathbf{f}},y}^k\| \leq \epsilon_{\overline{\mathbf{f}}}$*)*

Euclidean norms of residuals, $\|\mathbf{r}_{\overline{\mathbf{f}},x}^k\|$ and $\|\mathbf{r}_{\overline{\mathbf{f}},y}^k\|$, are sufficiently small (i.e., below a threshold $\epsilon_{\overline{\mathbf{f}}}$).

In the next section, we analyze the convergence of the Kaczmarz-type method for MT in Equation (37). A reconstruction algorithm for ultrasound tomography, the structure of which resembles the method developed for MT, was developed by [73], but its convergence was not discussed. The method for MT solves a specific type of nonlinear systems of equations by extending the Kaczmarz method [41, 42] and can be viewed as a special case of various Kaczmarz-type methods for nonlinear systems of equations (e.g., [54] and [58]).

3.3.1 Convergence Analysis

Given a nonlinear system of equations in Equation (31), we claim that the solution to the system of equations derived from by Equation (37), $\overline{\mathbf{f}}^k = \left(f_1^k, f_2^k, \cdots, f_P^k\right)$, $k = \{1, 2, \cdots\}$, converges to a true solution $\overline{\mathbf{f}}^\star = (f_1^\star, f_2^\star, \cdots, f_P^\star)$. For any matrix $\mathbf{A}(\mathbf{b})$

that is a function of a vector $\mathbf{b}$ with $\mathbf{A}_i(\mathbf{b})$ denoting its ith row, let us define $\mathbf{A}_i^+(\mathbf{b}) \equiv \frac{\mathbf{A}_i(\mathbf{b})^T}{\|\mathbf{A}_i(\mathbf{b})\|^2}$, referred to as the pseudoinverse of $\mathbf{A}_i(\mathbf{b})$ in this document.

Lemma 2. $\mathbf{A}_i^+(\mathbf{b})$ *satisfies the following four conditions for the Moore-Penrose pseudoinverse [67]:*

1. $\mathbf{A}_i(\mathbf{b})\mathbf{A}_i^+(\mathbf{b})\mathbf{A}_i(\mathbf{b}) = \mathbf{A}_i(\mathbf{b})$
2. $\mathbf{A}_i^+(\mathbf{b})\mathbf{A}_i(\mathbf{b})\mathbf{A}_i^+(\mathbf{b}) = \mathbf{A}_i^+(\mathbf{b})$
3. $(\mathbf{A}_i(\mathbf{b})\mathbf{A}_i^+(\mathbf{b}))^T = \mathbf{A}_i(\mathbf{b})\mathbf{A}_i^+(\mathbf{b})$
4. $(\mathbf{A}_i^+(\mathbf{b})\mathbf{A}_i(\mathbf{b}))^T = \mathbf{A}_i^+(\mathbf{b})\mathbf{A}_i(\mathbf{b})$

Proof. By simply substituting $\mathbf{A}_i^+(\mathbf{b})$ into the above four conditions, we can show that Lemma 2 holds. □

Before we prove the convergence of the Kaczmarz-type method for MT, we make the following two assumptions:

Assumption 4. $\mathbf{L}_i(\overline{\mathbf{f}})$ *in Equation* (31) *is Lipschitz continuous for all* $i = \{1, \cdots, K\}$ *with the largest Lipschitz constant* $\gamma_L = \max_i \gamma_L^i$*, where* γ_L^i *is the Lipschitz constant for* $\mathbf{L}_i(\overline{\mathbf{f}})$*.*

Assumption 4 implies that with a small variation of flow $\overline{\mathbf{f}}$, a corresponding variation in the time that individual vehicles spend in each grid cell is bounded, which is intuitively true.

Assumption 5. *Given any real numbers* $0 < \epsilon, \beta < 1$ *and a true solution* $\overline{\mathbf{f}}^\star$ *to Equation* (31)*, there exists a ball* $\mathcal{B}(\overline{\mathbf{f}}^\star, \delta)$ *around* $\overline{\mathbf{f}}^\star$ *with radius* $\delta > 0$ *such that the following hold for all* $\overline{\mathbf{f}} \in \mathcal{B}(\overline{\mathbf{f}}^\star, \delta)$*:*

1) $\dfrac{\gamma_L \|\overline{\mathbf{f}}^\star\|}{\|\mathbf{L}_i(\overline{\mathbf{f}})\|} < \sqrt{\epsilon}$ *for all* $i = \{1, \cdots, K\}$.

2) *For a sequence* $\overline{\mathbf{f}}^k$ *generated by Equation* (37)*, let* $\mathbf{e}_k = \overline{\mathbf{f}}^k - \overline{\mathbf{f}}^\star$ *and* $M_k = I - \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)$ *where* $i = \mod(k, K)+1$*. For every* K *iterations, there exists at least one* $k \in \{nK, nK+1, \cdots, (n+1)K-1\}$*,* $n = \{0, 1, 2, \cdots\}$ *such that* $\mathbf{e}_k$ *satisfies* $\langle M_k\mathbf{e}_k, \mathbf{e}_k\rangle \leq (1-\beta)\langle \mathbf{e}_k, \mathbf{e}_k\rangle$ *where* $\beta > 1 + \epsilon - \dfrac{1}{(1+\epsilon)^{K-1}}$.

In the following theorem, we prove the convergence of the method.

Theorem 2. *Suppose Assumptions 4 and 5 hold for Equation* (31) *and its solution* $\overline{\mathbf{f}}^\star$*. Starting from any initial point* $\overline{\mathbf{f}}^0$ *within a ball* $\mathfrak{B}(\overline{\mathbf{f}}^\star, \delta)$*, e.g.,* $\|\overline{\mathbf{f}}^\star - \overline{\mathbf{f}}^0\| < \delta$*, the sequence* $\overline{\mathbf{f}}^k$ *generated by Algorithm 1 converges to* $\overline{\mathbf{f}}^\star$ *as* $k \to \infty$*.*

Proof. Let us define an error term $\mathbf{e}_k = \overline{\mathbf{f}}^k - \overline{\mathbf{f}}^\star$. By subtracting $\overline{\mathbf{f}}^\star$ from the both sides of Equation (37) and substituting $\mathbf{L}_i^+(\overline{\mathbf{f}})$ and $r_k(\overline{\mathbf{f}}^k)$, we have

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \mathbf{L}_i^+(\overline{\mathbf{f}}^k)(\mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k - d_i), \tag{39}$$

where $i = \mod(k, K) + 1$. Since $d_i = \mathbf{L}_i(\overline{\mathbf{f}}^\star)\overline{\mathbf{f}}^\star$, we can express $\mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k - d_i$ as

$$\begin{aligned}
\mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k - d_i &= \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k - \mathbf{L}_i(\overline{\mathbf{f}}^\star)\overline{\mathbf{f}}^\star \\
&= \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k - \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^\star + \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^\star - \mathbf{L}_i(\overline{\mathbf{f}}^\star)\overline{\mathbf{f}}^\star \\
&= \mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k + \mathbf{p}_k\overline{\mathbf{f}}^\star,
\end{aligned} \tag{40}$$

where we define $\mathbf{p}_k = \mathbf{L}_i(\overline{\mathbf{f}}^k) - \mathbf{L}_i(\overline{\mathbf{f}}^\star)$. After substituting Equation (40) into Equation (39), the square of the Euclidean norm of the error is

$$\begin{aligned}
\langle \mathbf{e}_{k+1}, \mathbf{e}_{k+1}\rangle = \langle \mathbf{e}_k, \mathbf{e}_k\rangle &- 2\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k, \mathbf{e}_k\right\rangle - 2\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star, \mathbf{e}_k\right\rangle \\
&+ \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k\right\rangle + 2\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k\right\rangle \\
&+ \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star\right\rangle.
\end{aligned} \tag{41}$$

By the property of the inner product and Lemma 2, the fourth and fifth terms on the right side of Equation (41) become

$$\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k\right\rangle = \left\langle \left(\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)^T \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k, \mathbf{e}_k\right\rangle$$

$$= \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k, \mathbf{e}_k \right\rangle \tag{42}$$

and

$$\begin{aligned}\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\mathbf{e}_k \right\rangle &= \left\langle \left(\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)^T \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star, \mathbf{e}_k \right\rangle \\ &= \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star, \mathbf{e}_k \right\rangle, \end{aligned} \tag{43}$$

respectively. By substituting Equations (42) and (43) into Equation (41), we obtain

$$\begin{aligned}\langle \mathbf{e}_{k+1}, \mathbf{e}_{k+1} \rangle &= \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)\mathbf{e}_k, \mathbf{e}_k \right\rangle + \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star \right\rangle \\ &= \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)\mathbf{e}_k, \mathbf{e}_k \right\rangle + \left\| \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_k\overline{\mathbf{f}}^\star \right\|^2. \end{aligned} \tag{44}$$

Since $\mathbf{L}_i(\overline{\mathbf{f}})$ is Lipschitz continuous, we have

$$\begin{aligned}\|\mathbf{p}_k\| &= \|\mathbf{L}_i(\overline{\mathbf{f}}^k) - \mathbf{L}_i(\overline{\mathbf{f}}^\star)\| \\ &\leq \gamma_L \|\overline{\mathbf{f}}^k - \overline{\mathbf{f}}^\star\| \\ &= \gamma_L \|\mathbf{e}_k\|, \end{aligned}$$

in which γ_L is the Lipschitz constant in Assumption 4. Then, we obtain

$$\langle \mathbf{e}_{k+1}, \mathbf{e}_{k+1} \rangle \leq \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\right)\mathbf{e}_k, \mathbf{e}_k \right\rangle + \gamma_L^2 \|\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\|^2 \|\overline{\mathbf{f}}^\star\|^2 \langle \mathbf{e}_k, \mathbf{e}_k \rangle.$$

Let us define $M_k = I - \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)$. Since $\mathbf{L}_i^+(\overline{\mathbf{f}}^k) = \frac{\mathbf{L}_i(\overline{\mathbf{f}}^k)^T}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2}$, $\|\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\|^2 = \frac{1}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2}$, which gives

$$\begin{aligned}\|\mathbf{e}_{k+1}\|^2 &\leq \langle M_k\mathbf{e}_k, \mathbf{e}_k \rangle + \frac{\gamma_L^2 \|\overline{\mathbf{f}}^\star\|^2}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2} \langle \mathbf{e}_k, \mathbf{e}_k \rangle \\ &< \langle M_k\mathbf{e}_k, \mathbf{e}_k \rangle + \epsilon \langle \mathbf{e}_k, \mathbf{e}_k \rangle, \end{aligned}$$

where the last inequality is obtained by Assumption 5. By the definition of $\mathbf{L}_i^+(\overline{\mathbf{f}}^k)$, $\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)$ is normalized and it is a rank-one Hermitian matrix with the only non-zero eigenvalue being 1. Therefore, M_k is positive semidefinite with the largest eigenvalue being 1 (c.f., [37], Theorem 4.3.1 (Weyl)). By Assumption 5, there are the

following two cases:

Case 1.

$$\begin{aligned}\|\mathbf{e}_{k+1}\|^2 &< \langle \mathbf{e}_k, \mathbf{e}_k \rangle + \epsilon \langle \mathbf{e}_k, \mathbf{e}_k \rangle \\ &= (1+\epsilon) \|\mathbf{e}_k\|^2 .\end{aligned}$$

Case 2.

$$\begin{aligned}\|\mathbf{e}_{k+1}\|^2 &< (1-\beta) \langle \mathbf{e}_k, \mathbf{e}_k \rangle + \epsilon \langle \mathbf{e}_k, \mathbf{e}_k \rangle \\ &= (1-\beta+\epsilon) \|\mathbf{e}_k\|^2 .\end{aligned}$$

In the worst case, Case 2 happens only once and the rest corresponds to Case 1 for $k \in \{nK, nK+1, \cdots, (n+1)K-1\}$, $n = \{0, 1, 2, \cdots\}$. Without loss of generality, we can consider that the first instance over K is Case 2. Then, for $k = nK$, we have

$$\|\mathbf{e}_{k+1}\|^2 < (1-\beta+\epsilon) \|\mathbf{e}_k\|^2 .$$

At the next iteration, we obtain

$$\begin{aligned}\|\mathbf{e}_{k+2}\|^2 &< (1+\epsilon) \|\mathbf{e}_{k+1}\|^2 \\ &< (1+\epsilon)(1-\beta+\epsilon) \|\mathbf{e}_k\|^2 .\end{aligned}$$

By induction, we have

$$\|\mathbf{e}_{k+K}\|^2 < (1+\epsilon)^{K-1}(1-\beta+\epsilon) \|\mathbf{e}_k\|^2 . \tag{45}$$

By Assumption 5, Equation (45) becomes

$$\|\mathbf{e}_{k+K}\|^2 < \|\mathbf{e}_k\|^2 .$$

Therefore, $\mathbf{e}_k \to 0$ and $\overline{\mathbf{f}}^k \to \overline{\mathbf{f}}^\star$ as $k \to \infty$ in the L2 sense. This proves the theorem.

□

Remark 3. *Regardless of when Case 2 happens for* $k \in \{nK, nK+1, \cdots, (n+1)K-1\}$, $n = \{0, 1, 2, \cdots\}$, *we always have the same form of the equation as in Equation* (45). *If Case 2 happens* $1 < J \leq K$ *times, then*

$$\|\mathbf{e}_{k+K}\|^2 < (1+\epsilon)^{K-J}(1-\beta+\epsilon)^J \|\mathbf{e}_k\|^2 .$$

By Assumption 5, we have

$$\begin{aligned}\|\mathbf{e}_{k+K}\|^2 &< (1+\epsilon)^{K-J}\left(\frac{1}{(1+\epsilon)^{K-1}}\right)^J \|\mathbf{e}_k\|^2 \\ &= \frac{1}{(1+\epsilon)^{JK-K}} \|\mathbf{e}_k\|^2 .\end{aligned}$$

Since $1/(1+\epsilon)^{JK-K} < 1$, $\mathbf{e}_k \to 0$ *and* $\overline{\mathbf{f}}^k \to \overline{\mathbf{f}}^\star$ *as* $k \to \infty$ *in the L2 sense.*

3.4 Flow Field Mapping

Flow field mapping for MT iteratively runs trajectory tracing and flow field estimation and is described in Algorithm 2. Suppose we navigate K vehicles over observation interval $\mathcal{T} \in [t^0, t^f]$ and estimate a flow field using their motion-integration errors $\mathbf{d}_i$, $i = \{1, \cdots, K\}$. Let us use k to denote the index for iterations, $\mathbf{r}_i$ the actual positions of the vehicles, and $\tilde{\mathbf{r}}_i$ the predicted positions of the vehicles. First, we compute initial guesses of the flow field, $\overline{\mathbf{f}}^0$, and the vehicle trajectories, $\tilde{\gamma}_i^0$, between initial vehicle positions, $\mathbf{r}_{\gamma_i}(t^0)$, and final vehicle positions, $\mathbf{r}_{\gamma_i}(t^f)$ using Equation (1). Starting from $k = 1$, we compute the length of trajectories $L_{(i,j)}$ in each cell D_j and construct matrix $\mathbf{L}(\overline{\mathbf{f}})$ based on $\tilde{\gamma}_i^{k-1}$. Then, we estimate a flow field $\overline{\mathbf{f}}^k$ from $\overline{\mathbf{f}}^{k-1}$ by running Algorithm 1 and simulate new vehicle trajectories $\tilde{\gamma}_i^k$ based on $\overline{\mathbf{f}}^k$. To check the convergence of traced trajectories, we define an error term $\mathbf{e}_{\gamma^k} = \mathbf{r}_{\gamma_i}(t^f) - \mathbf{r}_{\tilde{\gamma}_i^k}(t^f)$ and compute the root-mean-square (rms) error $\mathbf{e}_{\gamma^k}^{rms}$. We repeat this process until the x and y components of $\Delta\mathbf{e}_{\gamma^k}^{rms} = \mathbf{e}_{\gamma^k}^{rms} - \mathbf{e}_{\gamma^{k-1}}^{rms}$ are sufficiently small (i.e., below a threshold ϵ_γ).

If a grid cell is not visited by a vehicle (e.g., for the jth cell for all i vehicles, $L_{(i,j)} = 0$ in Equation (21)), then there is no information from which to infer an underlying

Algorithm 2: MT flow field mapping

Data: Initial heading θ_i and motion-integration errors $\mathbf{d}_i$ of vehicles, $i = \{1, \cdots, K\}$

Set $k = 0$.
Compute an initial guess of the vehicle trajectories $\tilde{\gamma}_i^k$.
Compute an initial guess of the flow field $\overline{\mathbf{f}}^k$.
repeat
    Let $k = k + 1$.
    Construct matrix $\mathbf{L}(\overline{\mathbf{f}})$ based on the trajectories $\tilde{\gamma}_i^{k-1}$.
    Flow field estimation: Estimate the flow field $\overline{\mathbf{f}}^k$.
    Trajectory tracing: Simulate new trajectories $\tilde{\gamma}_i^k$ using the resulting flow field $\overline{\mathbf{f}}^k$.
    Compute the rms error $\mathbf{e}_{\gamma^k}^{rms}$ between simulated and real final positions.
until *a stopping condition is met (e.g.,* $|\Delta e_{x,\gamma^k}^{rms}|, |\Delta e_{y,\gamma^k}^{rms}| \leq \epsilon_\gamma$*)*

flow field for the cell. Therefore, we choose the grid size such that maximum number of cells are covered by vehicle trajectories. If vehicles are allowed to traverse the domain repeatedly, increasing the number of vehicle trajectories throughout the mapping domain will increase the spatial resolution and the accuracy, but it may require a longer observation period.

Since the system of equations in Equation (20) are underdetermined, the system may have infinitely many solutions. The initial guess to start the iterative estimation algorithm plays an important role to obtain a solution that is close to a true solution and may also affect the convergence rate. Inspired by the computation of glider-estimated flow (see, for details, [12]), we choose an initial guess as follows. For each vehicle, we first compute estimates of the flow velocity $\tilde{\mathbf{f}}_i$ over one observation interval such that

$$\tilde{\mathbf{f}}_i = \frac{\mathbf{r}_i(t^f) - \tilde{\mathbf{r}}_i(t^f)}{t^f - t^0}, \tag{46}$$

where $\mathbf{r}_i$ and $\tilde{\mathbf{r}}_i$ are the actual and predicted positions of the ith vehicle, respectively, and t^0 and t^f the starting and ending times of the observation, respectively. Then, for the grid cells that the trajectory of each vehicle passes through, we assign the

associated flow estimate. If multiple vehicles pass through a single cell, then we compute the average of flow velocities associated with the vehicles for the cell. Let us denote the set of vehicles that pass through the jth cell by $\mathfrak{S}_j$ and its cardinality $|\mathfrak{S}_j|$. Then, an initial guess for the estimation algorithm is given by

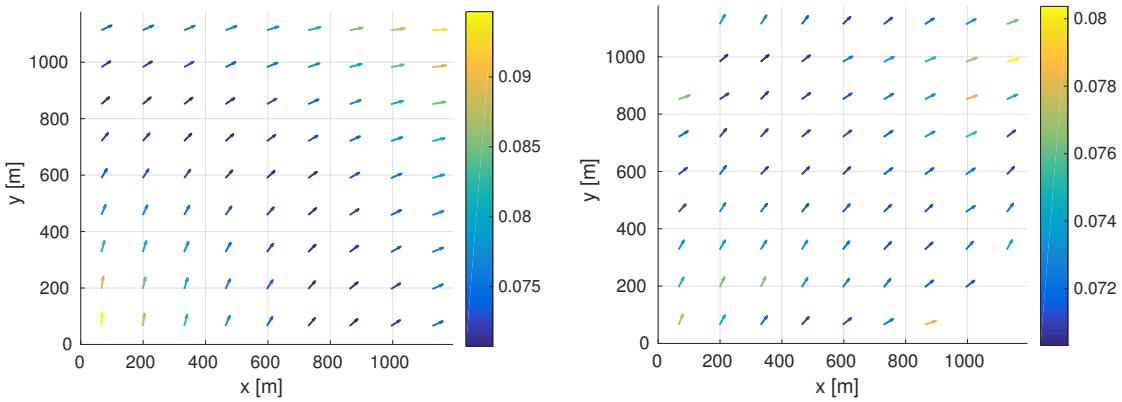
$$\overline{\mathbf{f}}^0 = \begin{bmatrix} \frac{1}{|\mathfrak{S}_1|} \sum_{i \in \mathfrak{S}_1} \tilde{\mathbf{f}}_i \\ \vdots \\ \frac{1}{|\mathfrak{S}_P|} \sum_{i \in \mathfrak{S}_P} \tilde{\mathbf{f}}_i \end{bmatrix}. \tag{47}$$

3.5 Simulation and Experimental Results

This section validates MT through both simulations and experiments. We have collected the following data sets: 1) simulations using UMSAs under a simulated flow field, 2) indoor experiments using mobile robots that mimic the motion of UMSAs under a simulated flow field, and 3) a field experiment using a glider deployed off the coast of Georgia in September 2013. From the collected data sets, the proposed algorithms estimate vehicle trajectories and flow fields. In the flow field estimation algorithm in Algorithm 1, we use $\lambda^k = 0.01$, $\forall k$. To determine the convergence for trajectory tracing and flow field estimation, we empirically choose small thresholds, $\epsilon_{\overline{\mathbf{f}}}$, and ϵ_γ in Algorithms 1 and 2, respectively. Let us denote the true flow field by $\overline{\mathbf{f}}^\star$ and the estimated flow field map by $\overline{\mathbf{f}}$. If true flow field $\overline{\mathbf{f}}^\star$ is available, we define estimation error as $\mathbf{e} = \overline{\mathbf{f}}^\star - \overline{\mathbf{f}}$. The performance of flow field mapping is evaluated by computing the rms of the errors in the x and y components, and while discussing the results from data sets 1 through 3, we show how MT performs when the assumptions on the vehicle speed and the flow field are violated.

3.5.1 Simulations

First, we test MT with the ideal case (e.g., identical simulated vehicles in a smooth time-invariant field) using synthetic data generated by simulated UMSAs under a



(a) A simulated "true" flow field. (b) A constructed non-parametric flow field.

Figure 7: A simulated "true" flow field and a constructed flow field from UMSA simulation data.

simulated field constructed in a $1200\,m \times 1200\,m$ domain. We simulate a flow field such that as the distance from the current location to the origin increases, the x component of the flow velocity increases from 0 to $0.1\,m/s$ and the y component of the flow velocity decreases from $0.1\,m/s$ to 0. Figure 7a shows the "true" flow field $\overline{\mathbf{f}}^{\star}$. In this simulated flow field, we deploy $K = 18$ vehicles to move straight forward at the horizontal through-water speed of 0.35 m/s for one observation interval. Of the 18 vehicles, nine travel from the left of the domain to the right and the other nine from the bottom to the top. The starting, target, and final positions, and the real and predicted trajectories of the vehicles are illustrated in Figure 8.

Given a waypoint $\mathbf{w}_i$, the ith vehicle first computes heading θ_i towards the waypoint and moves towards it according to Equation (1). That is, for all $i = \{1, \cdots, K\}$, the predicted trajectory is computed as $\tilde{\mathbf{r}}_i(k+1) = \tilde{\mathbf{r}}_i(k) + s_{\mathrm{h}}[\cos\theta_i, \sin\theta_i]^T \Delta t$, in which k is the discrete time step and Δt is the step size (in this simulation, $\Delta t = 1$). The dynamics of the actual trajectory follow Equation (2), and the actual trajectory is formed by computing $\mathbf{r}_i(k+1) = \mathbf{r}_i(k) + s_{\mathrm{h}}[\cos\theta_i, \sin\theta_i]^T \Delta t + \mathbf{f}(\mathbf{r}_i)\Delta t$. A UMSA stops its navigation when a stopping condition is met, e.g., $\|\mathbf{w}_i - \tilde{\mathbf{r}}_i\| < 1\,m$. Once all the UMSAs finish their navigation, we compute their motion-integration errors $\mathbf{d}_i$. Then,

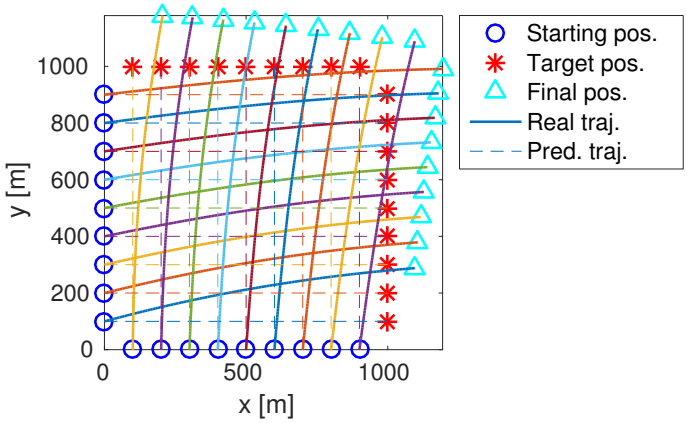


Figure 8: UMSA simulation. The starting positions (the blue circles), target positions (the red stars), and final positions (the cyan triangles) of 18 UMSAs are displayed. Solid lines connecting starting positions and final positions are real vehicle trajectories, and dashed lines connecting starting positions and target positions are predicted trajectories.

we trace vehicle trajectories $\tilde{\gamma}_i$ and, using the resulting traced vehicle trajectories, estimate flow field $\overline{\mathbf{f}}$.

Figure 7b displays a constructed flow field and Table 1 shows convergence conditions for the algorithms and mapping errors. Having 9 vehicles navigating horizontally and 9 vertically (i.e., 18 measurements), the flow field is constructed in a 9×9 grid domain (i.e., 81 unknowns in each system of equations in Equation (20)). As Figure 8 shows that the actual vehicle trajectories do not cover the regions at the left upper and right bottom corners of the domain, the estimation algorithm is not able to construct flow in these regions. Therefore, the regions where flow is not estimated are omitted from the error computation. Considering the mapping errors, although the dependency of MT on the vehicle trajectory is not trivial, MT successfully constructs a flow field map from simulated vehicle data with high fidelity in the ideal case.

3.5.2 Indoor experiments

Next, MT is applied to the data collected from mobile robots in a simulated time-invariant field through indoor experiments using Khepera III mobile robots. The

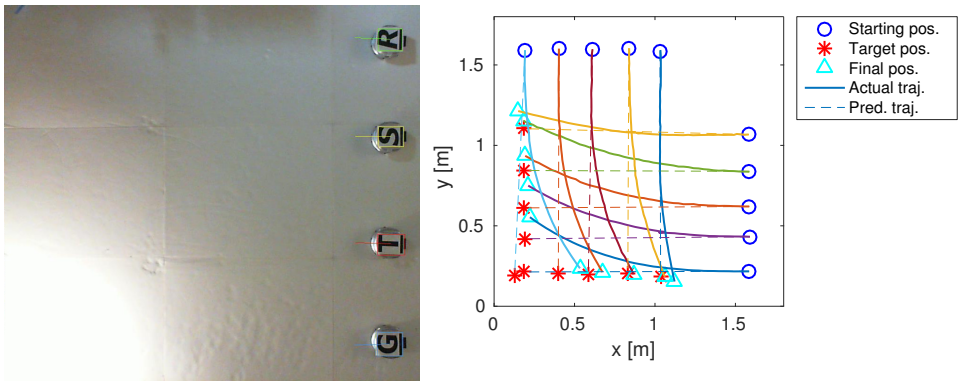
Table 1: Convergence conditions and quantitative results for non-parametric MT simulations.

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$
$\epsilon_\gamma = 10^{-2}$	$\epsilon_{\overline{\mathbf{f}}} = 30$	$x : 0.014456\, m/s$ $y : 0.014457\, m/s$

motor of the robot is controlled by pulse signals, and one pulse signal sent to the motors rotates the robot by 0.06°, so the rotation angle is a multiple of 0.06°. In addition, even with an identical command given to all the robots used in the experiment, the motions of the robots are all different. Because of these limitations, Assumption 2 (constant and identical speed for all vehicles) is violated.

The experimental setup in a $1600\, m \times 1600\, m$ domain is shown in Figure 9a. To construct a simulated flow field in the domain, we place a light source at the left bottom corner $(x, y) = (0, 0)$ of the domain and simulate a flow field using ambient light intensity, which is measured by 9 infra-red (IR) sensors located on the side of each robot. At each location of a robot, we compute the mean of the light intensity measurements collected from the 9 IR sensors of the robot. Because light intensity measurements range from 0 to $I_{\max} = 4096$, where a lower value indicates higher light intensity, we reverse the range of the measurements by subtracting the mean of the measurements at each location from I_{max} and compute a ratio of the mean light intensity to I_{max}. Then, we scale the ratio by scaling factor $c = \frac{3\sqrt{2}}{10}$. The simulated flow field is constructed by multiplying the scaled ratio by direction vector $[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$. The simulated "true" flow field, $\overline{\mathbf{f}}^\star$, shown in Figure 10a is obtained separately from MT mapping experiments by densely sampling light intensity throughout the domain using a Khepera III robot. The scaled ratio of this dense light intensity measurement is multiplied by the direction vector to construct the simulated "true" flow field.

To estimate a flow field map through MT, Khepera III robots traverse the domain. As is the case with the AUV simulation in Section 3.5.1, given an initial heading θ_i

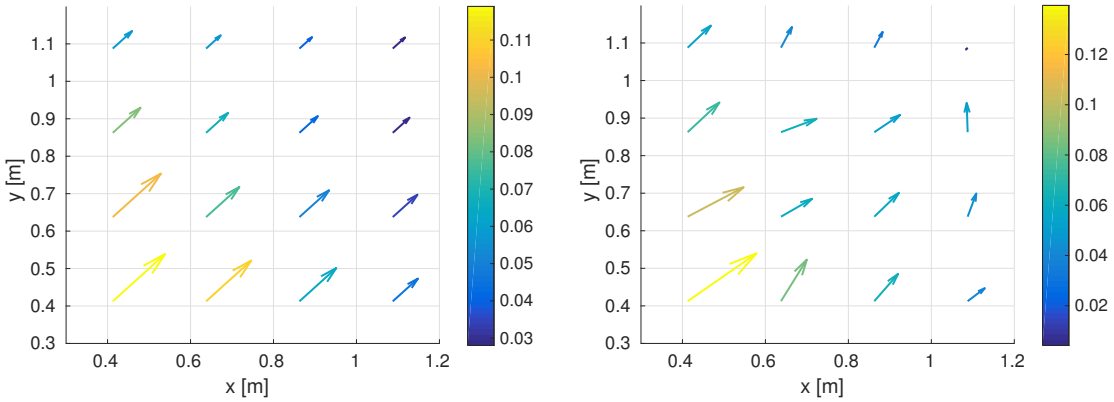


(a) Four Khepera III robots in a light field. (b) Trajectories of Khepera III robots.

Figure 9: 9a shows experimental setup with Khepera III robots in a light field. Four Khepera III robots are differentiated using letters 'G','T','S', and 'R'. National Instruments LabVIEW identifies the positions (the colored rectangle around each robot) and headings (the line attached to each robot) of the robots. A light source is located at the left bottom corner to simulate a flow field. 9b displays trajectories of Khepera III robots. The green circles, red stars, and cyan triangles represent the starting positions, target positions, and final positions of the robots, respectively. Solid lines connecting starting positions and final positions are real trajectories, and dashed lines connecting starting positions and target positions are planned trajectories.

of the ith robot, the robots move towards their target with time step $\Delta t = 0.1$. At each location of a robot, it measures ambient light intensity in the domain with a light source and sends the measurement to a central computer that computes flow velocity at the location of the robot in the simulated flow field. Then, with this flow information, the robots are controlled to follow the horizontal motion of AUVs represented by Equations (1) and (2). Even though the actual trajectory of a robot is observable through a camera installed on top of the experimental domain, we treat the actual trajectory of a robot as unknown since the underwater trajectory is unknown in AUV operations. Only the starting and final positions of robots are used for flow field mapping. Once all the robots finish their travel, trajectory information is collected on the central computer for flow field mapping through MT.

We ran multiple sets of experiment using four Khepera III robots moving straight forward by $1.4\,m$ with horizontal speed $s_{\mathrm{h}} = 0.3\,m/s$ and chose a set of $K = 10$



(a) A simulated "true" flow field.

(b) A constructed non-parametric flow field.

Figure 10: A simulated "true" flow field and a constructed flow field from indoor experimental data using Khepera robots under simulated flow.

trajectories of the robots – five from the right side of the domain to the left and five from the top to the bottom – shown in Figure 9b. Figure 10b shows the constructed flow field and Table 2 shows convergence conditions for the algorithms and mapping errors. Even though we have 5 horizontal trajectories and 5 vertical trajectories, since the experiments using Khepera III robots violate the assumption of the constant and identical vehicle speed, we discretize the flow field into 4×4 for the sub-domain ($[0.3, 1.2]\,m \times [0.3, 1.2]\,m$) to increase the mapping accuracy. Compared to the magnitude range of the true flow field ([0.03,0.12]), the magnitude range of the estimated flow field ([0,0.14]) is slightly enlarged. However, the average of the estimated flow field (0.0392 m/s in x and 0.0411 m/s in y) is very close to that of the true flow field (0.0442 m/s in x and 0.0442 m/s in y). In addition, Table 2 shows small mapping errors. These results show that under the violation of Assumption 2, MT can provide the knowledge of flow fields with sufficient accuracy.

3.5.3 A field experiment

Lastly, we validate MT using real experimental data collected by an underwater glider in a time-varying flow field. Because of its limited localization capabilities, the

Table 2: Convergence conditions and quantitative results for non-parametric MT indoor experiments.

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$
$\epsilon_\gamma = 10^{-4}$	$\epsilon_{\bar{\mathbf{f}}} = 0.51$	$x : 0.018154\, m/s$ $y : 0.014197\, m/s$

glider navigates via dead-reckoning, which estimates the position of the glider using estimates of glider speed, compass heading, and flow velocity. While navigating, the glider moves in saw-tooth underwater patterns by repeating dive and climb motions, sampling most of the water column along the glider trajectory. The glider regularly comes to the surface of water, operationally every 4-6 $hours$, for GPS updates and data transfers to an on-shore computer. Because of the influence of flow, when the glider surfaces, we typically observe the motion-integration error, also known as the dead-reckoning error for gliders.

Upon surfacing, a glider computes an estimate of average flow velocity along the trajectory over one subsurface phase based on the motion-integration error (see Equation (46)). This glider-derived flow estimate can be either incorporated into navigation (i.e., $\tilde{\mathbf{f}} \neq \mathbf{0}$ in Equation (1)) to reduce the motion-integration error or deactivated (i.e., $\tilde{\mathbf{f}} = \mathbf{0}$) so that no flow estimate is used in navigation. Then, the glider-derived flow estimate at a surfacing event is computed by adding the new flow estimate to the previous flow estimate that is used to navigate over the previous subsurface phase.

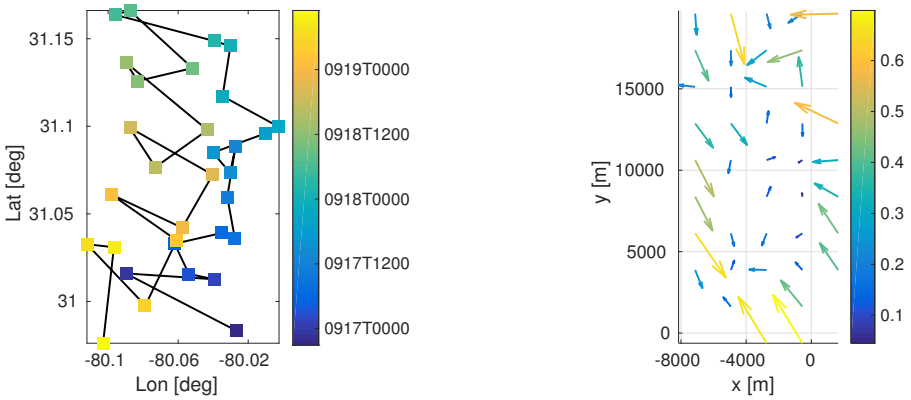
As part of a collaborative effort between the Georgia Institute of Technology and the Skidaway Institute of Oceanography, we participated in Gliderpalooza 2013 [18], a coordinated deployment of gliders along the US East Coast, in September 2013. During Gliderpalooza 2013, 17 gliders were simultaneously deployed in the east coast of North America, and glider data were used to improve the accuracy of existing ocean models through data assimilation and to study fish and marine mammal migrations.

We deployed one glider (*Modena*) off the coast of Georgia, and Modena navigated with 4-6 $hours$ surfacing intervals along a triangle track that includes one edge aligned along the edge of the Gulf Stream.

For flow field mapping, we selected 31 consecutive segments of the glider trajectories (see Figure 11a) from 18:00 September 16 to 22:00 September 21, 2013 and treated these segments as the trajectories of $K = 31$ gliders. The glider was moving north at approximately 0.275 m/s on average next to the edge of the Gulf Stream based on a navigation method developed in [12] using a predictive ocean model. During this period, the maximum flow speed estimated by the glider was 0.5687 m/s, the minimum 0.0899 m/s, and the average 0.3416 m/s, which is extremely dynamic for the glider and significantly influences the vehicle trajectories. Modena was pushed towards the south and west by strong currents in excess of the vehicles forward speed.

Figure 11b shows the constructed flow field in a 10×5 grid domain using MT after $20 \times K$ iterations. Since existing ocean observing systems and models available to us do not provide reliable data for the deployment time and area, the true flow field is unknown. Therefore, the convergence conditions cannot be empirically chosen and flow field mapping error cannot be computed. The glider traveled under a time-varying flow field, so we use glider trajectories estimated on the on-board glider system, instead of using the trajectory tracing technique presented in this document. The on-board estimated glider trajectories are based on a dead-reckoning method that integrates the vehicle position based on the heading angle measured from a compass and the vehicle speed estimated from the rate of change of the depth. Even though the speed of a glider is often considered constant, it varies while the glider performs dive-climb cycles and is also affected by water depth. Therefore, both Assumptions 1 and 2 are not satisfied.

Even though the assumptions are violated, the results demonstrate that MT is still able to construct maps of flow fields from the experimental data. The resulting



(a) Glider trajectories.

(b) A constructed non-parametric flow field.

Figure 11: 11a Glider trajectories with surfacing positions during the Gliderpalooza 2013 deployment from 18:00 September 16 to 22:00 September 21, 2013. 11b A constructed non-parametric flow field from the experimental data.

flow field map through MT is largely affected by time-varying flow along the vehicle trajectories. In the boundary grid cells of the domain where northwards and southwards trajectories do not overlap, flows exhibit the same pattern that represents the actual trajectories that wind up around the boundary from the right bottom corner of the domain through the top to the left bottom corner. In the inner cells of the domain where northwards and southwards trajectories overlap, we observe weaker flows. We analyze that this result is caused by the strategy of initial flow computation introduced in Section 3.4. As discussed in Section 3.4, if multiple trajectories overlap within a cell, we compute the initial guess of flow for the cell by averaging estimated flow values for the trajectories, and thus we see that the mapping results are largely affected by the initial guess of the flow field.

CHAPTER IV

PARAMETRIC MOTION TOMOGRAPHY

This section presents a parametric version of MT that incorporates a data-driven flow model. By approximating a flow field using the basis functions and corresponding parameters of a data-driven flow model, parametric MT has a less dimensional solution space than non-parametric MT and may reduce noise of the estimated flow field for a smooth field. By solving the parametric MT problem, parameters of a data-driven flow model are estimated from the motion-integration error and trajectory information of UMSAs.

4.1 Formulation of Parametric MT

As introduced in Section 2.2.1, flow can be represented by a parameterized expression using a data-driven flow model (Equation (4) or Equation (5)). Since we deal with the time-invariant flow field yet, only spatial basis functions are required to model the flow. Consider an $R \times S$ grid domain with $\mathbf{f}_j \in \mathbb{R}^2$, $j = \{1, \cdots, P = R \times S\}$ denoting flow velocity for the jth grid cell. Based on the data-driven flow model, $\mathbf{f}_j$ can be written as

$$\mathbf{f}_j = \sum_{m=1}^{M} \eta_m \phi_m(\mathbf{r}_j), \tag{48}$$

where $\eta_m \in \mathbb{R}^2$ are spatial parameter vectors, $\phi_m \in \mathbb{R}$ are spatial basis functions, M is the number of spatial basis functions, and $\mathbf{r}_j \in \mathbb{R}^2$ is the position that represents the jth cell (e.g., the center of the cell).

Now, consider flow velocity along the x and y directions separately. We denote a set of parameters in x and y components by $\mathbf{h}_x = [\eta_{x,1}, \eta_{x,2}, \cdots, \eta_{x,M}]^T$ and $\mathbf{h}_y = [\eta_{y,1}, \eta_{y,2}, \cdots, \eta_{y,M}]^T$, respectively. We define matrix $\mathbf{H}$ such that its jth row

is defined as $[\phi_1(\mathbf{r}_j), \cdots, \phi_M(\mathbf{r}_j)]$, $j = \{1, \cdots, P\}$. Then, $\overline{\mathbf{f}}_x = [f_{x,1}, \cdots, f_{x,P}]^T$ and $\overline{\mathbf{f}}_y = [f_{y,1}, \cdots, f_{y,P}]^T$ are represented by

$$\begin{aligned} \overline{\mathbf{f}}_x &= \mathbf{H}\mathbf{h}_x \\ \overline{\mathbf{f}}_y &= \mathbf{H}\mathbf{h}_y. \end{aligned} \tag{49}$$

By substituting Equation (49) into Equation (20), the motion-integration error equation can be parameterized as

$$\begin{aligned} \mathbf{d}_x &= \mathbf{L}(\mathbf{h})\mathbf{H}\mathbf{h}_x \\ \mathbf{d}_y &= \mathbf{L}(\mathbf{h})\mathbf{H}\mathbf{h}_y, \end{aligned} \tag{50}$$

where $\mathbf{h} = [\mathbf{h}_x^T, \mathbf{h}_y^T]^T$. By defining $\mathbf{G}(\mathbf{h}) = \mathbf{L}(\mathbf{H}\mathbf{h})\mathbf{H}$, the parameterized motion-integration error equation can be rewritten as

$$\begin{aligned} \mathbf{d}_x &= \mathbf{G}(\mathbf{h})\mathbf{h}_x \\ \mathbf{d}_y &= \mathbf{G}(\mathbf{h})\mathbf{h}_y. \end{aligned} \tag{51}$$

As is the case with non-parametric MT in Chapter 3, we solve the parametric motion-integration error in Equation (51) for parameters $\mathbf{h}$ to estimate flow $\overline{\mathbf{f}}$ in Equation (49).

4.2 A Kaczmarz-Type Method for Flow Field Estimation by Parametric MT

Let us omit x and y in the system for simplicity of presentation, i.e., we have

$$\mathbf{d} = \mathbf{G}(\mathbf{h})\mathbf{h}, \tag{52}$$

where

$$\mathbf{G}(\mathbf{h}) = \begin{bmatrix} \mathbf{G}_1(\mathbf{h}) \\ \vdots \\ \mathbf{G}_K(\mathbf{h}) \end{bmatrix}, \ \mathbf{h} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_M \end{bmatrix}, \ \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix}.$$

We solve equations in the form of Equation (52) for $\mathbf{h}$ by iterating the following optimization process:

$$\begin{aligned} \mathbf{h}^{k+1} &= \underset{\mathbf{h}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathbf{h} - \mathbf{h}^k \right\|^2 \\ &\text{subject to} \quad d_i = \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}. \end{aligned} \tag{53}$$

This optimization problem can be solved by the Lagrange multiplier method such that

$$\mathbf{h}^{k+1} = \underset{\mathbf{h}}{\operatorname{argmin}} \left[\frac{1}{2} \left(\left\| \mathbf{h} - \mathbf{h}^k \right\|^2 \right) + \lambda \left(d_i - \mathbf{G}_i(\mathbf{h}^k)\mathbf{h} \right) \right],$$

where λ is the Lagrange multiplier and leads to

$$\mathbf{h}^{k+1} = \mathbf{h}^k + \frac{d_i - \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^k}{\|\mathbf{G}_i(\mathbf{h}^k)\|^2} \left(\mathbf{G}_i(\mathbf{h}^k) \right)^T, \tag{54}$$

where $\mathbf{G}_i$ is the ith row of matrix $\mathbf{G}$ and i satisfies $i = \text{mod}(k, K) + 1$.

As is the case with the non-parametric flow field algorithm, we process every K iterations from $i = 1$ to K as a batch, so upon convergence, the iteration number is always a multiple of K. Based on Equation (54), we derive a parametric flow field estimation algorithm (Algorithm 3) with relaxation parameter λ^k. To check the convergence, for k that satisfies $\text{mod}(k, K) + 1 = K$, we define residuals $\mathbf{r}^k_{\mathbf{h},x} = \mathbf{G}(\mathbf{h}^k)\mathbf{h}^k_x - \mathbf{d}_x$ and $\mathbf{r}^k_{\mathbf{h},y} = \mathbf{G}(\mathbf{h}^k)\mathbf{h}^k_y - \mathbf{d}_y$. The algorithm runs until $\|\mathbf{r}^k_{\mathbf{h},x}\|$ and $\|\mathbf{r}^k_{\mathbf{h},y}\|$ are sufficiently small (i.e., below a threshold $\epsilon_{\mathbf{h}}$). For parametric flow field mapping through MT, we replace Algorithm 1 in Algorithm 2 with Algorithm 3.

4.2.1 Convergence Analysis

The convergence proof for the Kaczmarz-type method for parametric MT in Equation (54) follows the same procedure as that of the method for non-parametric MT in Section 3.3.1. Consider a nonlinear system of equations in Equation (52). We claim that the solution to the system of equations derived by Equation (54), $\mathbf{h}^k = \left(\eta^k_1, \eta^k_2, \cdots, \eta^k_M\right)$, $k = \{1, 2, \cdots\}$, converges to a true solution $\mathbf{h}^\star = (\eta^\star_1, \eta^\star_2, \cdots, \eta^\star_M)$. We make the following two assumptions:

Algorithm 3: Parametric MT flow field estimation

Data: Motion-integration errors $\mathbf{d}_i, i = \{1, \cdots, K\}$

Set $k = 0$. Make an initial guess of the solutions, $\mathbf{h}_x^0$ and $\mathbf{h}_y^0$.

repeat

  for $i = 1$ **to** K **do**

    Update the solutions by

$$\mathbf{h}_x^{k+1} = \mathbf{h}_x^k + \lambda^k \frac{d_{x,i} - \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}_x^k}{\|\mathbf{G}_i(\mathbf{h}^k)\|^2} \left(\mathbf{G}_i(\mathbf{h}^k)\right)^T,$$

$$\mathbf{h}_y^{k+1} = \mathbf{h}_y^k + \lambda^k \frac{d_{y,i} - \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}_y^k}{\|\mathbf{G}_i(\mathbf{h}^k)\|^2} \left(\mathbf{G}_i(\mathbf{h}^k)\right)^T.$$

    Let $k = k + 1$.

  end

  $\mathbf{r}_{\mathbf{h},x}^k = \mathbf{G}(\mathbf{h}^k)\mathbf{h}_x^k - \mathbf{d}_x$.

  $\mathbf{r}_{\mathbf{h},y}^k = \mathbf{G}(\mathbf{h}^k)\mathbf{h}_y^k - \mathbf{d}_y$.

until *a stopping condition is met (e.g.,* $\|\mathbf{r}_{\mathbf{h},x}^k\|, \|\mathbf{r}_{\mathbf{h},y}^k\| \leq \epsilon_{\mathbf{h}}$*)*

Assumption 6. $\mathbf{G}_i(\mathbf{h})$ *in Equation* (52) *is Lipschitz continuous for all* $i = \{1, \cdots, K\}$ *with the largest Lipschitz constant* $\gamma_G = \max_i \gamma_G^i$*, where* γ_G^i *is the Lipschitz constant for* $\mathbf{G}_i(\mathbf{h})$*.*

Assumption 7. *Given any real numbers* $0 < \epsilon, \beta < 1$ *and a true solution* $\mathbf{h}^\star$ *to Equation* (52)*, there exists a ball* $\mathfrak{B}(\mathbf{h}^\star, \delta)$ *around* $\mathbf{h}^\star$ *with radius* $\delta > 0$ *such that the following hold for all* $\mathbf{h} \in \mathfrak{B}(\mathbf{h}^\star, \delta)$*:*

1) $\dfrac{\gamma_G \|\mathbf{h}^\star\|}{\|\mathbf{G}_i(\mathbf{h})\|} < \sqrt{\epsilon}$ *for all* $i = \{1, \cdots, K\}$*.*

2) *For a sequence* $\mathbf{h}^k$ *generated by Equation* (54)*, let* $\mathbf{e}_k = \mathbf{h}^k - \mathbf{h}^\star$ *and* $M_k = I - \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)$ *where* $i = \mathrm{mod}(k, K) + 1$*. For every* K *iterations, there exists at least one* $k \in \{nK, nK + 1, \cdots, (n+1)K - 1\}$*,* $n = \{0, 1, 2, \cdots\}$ *such that* $\mathbf{e}_k$ *satisfies* $\langle M_k \mathbf{e}_k, \mathbf{e}_k \rangle \leq (1 - \beta) \langle \mathbf{e}_k, \mathbf{e}_k \rangle$ *where* $\beta > 1 + \epsilon - \dfrac{1}{(1+\epsilon)^{K-1}}$*.*

In the following theorem, we prove the convergence of the method.

Theorem 3. *Suppose Assumptions 6 and 7 hold for Equation* (52) *and its solution* $\mathbf{h}^\star$*. Starting from any initial point* $\mathbf{h}^0$ *within a ball* $\mathfrak{B}(\mathbf{h}^\star, \delta)$*, e.g.,* $\mathrm{dist}(\mathbf{h}^\star, \mathbf{h}^0) < \delta$*,*

the sequence $\mathbf{h}^k$ *generated by Algorithm 3 converges to* $\mathbf{h}^\star$ *as* $k \to \infty$.

Proof. Let us define an error term $\mathbf{e}_k = \mathbf{h}^k - \mathbf{h}^\star$. By subtracting $\mathbf{h}^\star$ from the both sides of Equation (54) and substituting $\mathbf{G}_i^+(\mathbf{h})$, we have

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \mathbf{G}_i^+(\mathbf{h}^k)(\mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^k - d_i), \tag{55}$$

where $i = \mathrm{mod}(k, K) + 1$. Since $d_i = \mathbf{G}_i(\mathbf{h}^\star)\mathbf{h}^\star$, we can express $\mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^k - d_i$ as

$$\begin{aligned}
\mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^k - d_i &= \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^k - \mathbf{G}_i(\mathbf{h}^\star)\mathbf{h}^\star \\
&= \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^k - \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^\star + \mathbf{G}_i(\mathbf{h}^k)\mathbf{h}^\star - \mathbf{G}_i(\mathbf{h}^\star)\mathbf{h}^\star \\
&= \mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k + \mathbf{p}_k\mathbf{h}^\star,
\end{aligned} \tag{56}$$

where we define $\mathbf{p}_k = \mathbf{G}_i(\mathbf{h}^k) - \mathbf{G}_i(\mathbf{h}^\star)$. After substituting Equation (56) into Equation (55), the square of the Euclidean norm of the error is

$$\begin{aligned}
\langle \mathbf{e}_{k+1}, \mathbf{e}_{k+1} \rangle = \langle \mathbf{e}_k, \mathbf{e}_k \rangle &- 2\left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k, \mathbf{e}_k \right\rangle - 2\left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star, \mathbf{e}_k \right\rangle \\
&+ \left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k, \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k \right\rangle + 2\left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star, \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k \right\rangle \\
&+ \left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star, \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star \right\rangle.
\end{aligned} \tag{57}$$

By the property of the inner product and Lemma 2, the fourth and fifth terms on the right side of Equation (57) become

$$\begin{aligned}
\left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k, \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k \right\rangle &= \left\langle \left(\mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\right)^T \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k, \mathbf{e}_k \right\rangle \\
&= \left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k, \mathbf{e}_k \right\rangle
\end{aligned} \tag{58}$$

and

$$\begin{aligned}
\left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star, \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\mathbf{e}_k \right\rangle &= \left\langle \left(\mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\right)^T \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star, \mathbf{e}_k \right\rangle \\
&= \left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star, \mathbf{e}_k \right\rangle,
\end{aligned} \tag{59}$$

respectively. By substituting Equations (58) and (59) into Equation (57), we have

$$\begin{aligned}
\langle \mathbf{e}_{k+1}, \mathbf{e}_{k+1} \rangle &= \left\langle \left(I - \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\right)\mathbf{e}_k, \mathbf{e}_k \right\rangle + \left\langle \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star, \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star \right\rangle \\
&= \left\langle \left(I - \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\right)\mathbf{e}_k, \mathbf{e}_k \right\rangle + \left\| \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{p}_k\mathbf{h}^\star \right\|^2.
\end{aligned} \tag{60}$$

Since $\mathbf{G}_i(\mathbf{h})$ is Lipschitz continuous,

$$\begin{aligned}\|\mathbf{p}_k\| &= \|\mathbf{G}_i(\mathbf{h}^k) - \mathbf{G}_i(\mathbf{h}^\star)\| \\ &\leq \gamma_L \|\mathbf{h}^k - \mathbf{h}^\star\| \\ &= \gamma_L \|\mathbf{e}_k\|,\end{aligned}$$

in which γ_L is the Lipschitz constant in Assumption 6. Then, we have

$$\langle \mathbf{e}_{k+1}, \mathbf{e}_{k+1} \rangle \leq \left\langle \left(I - \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)\right) \mathbf{e}_k, \mathbf{e}_k \right\rangle + \gamma_L^2 \|\mathbf{G}_i^+(\mathbf{h}^k)\|^2 \|\mathbf{h}^\star\|^2 \langle \mathbf{e}_k, \mathbf{e}_k \rangle .$$

Let us define $M_k = I - \mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)$. Since $\mathbf{G}_i^+(\mathbf{h}^k) = \frac{\mathbf{G}_i(\mathbf{h}^k)^T}{\|\mathbf{G}_i(\mathbf{h}^k)\|^2}$, $\|\mathbf{G}_i^+(\mathbf{h}^k)\|^2 = \frac{1}{\|\mathbf{G}_i(\mathbf{h}^k)\|^2}$, which gives

$$\begin{aligned}\|\mathbf{e}_{k+1}\|^2 &\leq \langle M_k \mathbf{e}_k, \mathbf{e}_k \rangle + \frac{\gamma_L^2 \|\mathbf{h}^\star\|^2}{\|\mathbf{G}_i(\mathbf{h}^k)\|^2} \langle \mathbf{e}_k, \mathbf{e}_k \rangle \\ &< \langle M_k \mathbf{e}_k, \mathbf{e}_k \rangle + \epsilon \langle \mathbf{e}_k, \mathbf{e}_k \rangle ,\end{aligned}$$

where the last inequality is obtained by Assumption 7. By the definition of $\mathbf{G}_i^+(\mathbf{h}^k)$, $\mathbf{G}_i^+(\mathbf{h}^k)\mathbf{G}_i(\mathbf{h}^k)$ is normalized and it is a rank-one Hermitian matrix with the only non-zero eigenvalue being 1. Therefore, M_k is positive semidefinite with the largest eigenvalue being 1 (c.f., [37], Theorem 4.3.1 (Weyl)) By Assumption 7, there are the following two cases:

Case 1.

$$\begin{aligned}\|\mathbf{e}_{k+1}\|^2 &< \langle \mathbf{e}_k, \mathbf{e}_k \rangle + \epsilon \langle \mathbf{e}_k, \mathbf{e}_k \rangle \\ &= (1+\epsilon) \|\mathbf{e}_k\|^2 .\end{aligned}$$

Case 2.

$$\begin{aligned}\|\mathbf{e}_{k+1}\|^2 &< (1-\beta) \langle \mathbf{e}_k, \mathbf{e}_k \rangle + \epsilon \langle \mathbf{e}_k, \mathbf{e}_k \rangle \\ &= (1-\beta+\epsilon) \|\mathbf{e}_k\|^2 .\end{aligned}$$

In the worst case, Case 2 happens only once and the rest corresponds to Case 1 for $k \in \{nK, nK+1, \cdots, (n+1)K-1\}$, $n = \{0, 1, 2, \cdots\}$. Without loss of generality, we

can consider that the first instance over K iterations is Case 2. Then, for $k = nK$, we have

$$\|\mathbf{e}_{k+1}\|^2 < (1 - \beta + \epsilon) \|\mathbf{e}_k\|^2 .$$

At the next iteration, we obtain

$$\begin{aligned} \|\mathbf{e}_{k+2}\|^2 &< (1 + \epsilon) \|\mathbf{e}_{k+1}\|^2 \\ &< (1 + \epsilon)(1 - \beta + \epsilon) \|\mathbf{e}_k\|^2 . \end{aligned}$$

By induction, we have

$$\|\mathbf{e}_{k+K}\|^2 < (1 + \epsilon)^{K-1}(1 - \beta + \epsilon) \|\mathbf{e}_k\|^2 . \tag{61}$$

By Assumption 7, Equation (61) becomes

$$\|\mathbf{e}_{k+K}\|^2 < \|\mathbf{e}_k\|^2 .$$

Therefore, $\mathbf{e}_k \to 0$ and $\mathbf{h}^k \to \mathbf{h}^\star$ as $k \to \infty$ in the L2 sense. This proves the theorem. □

4.3 Spatial Basis Functions for Parametric MT

Radial basis functions (RBFs) are known to have properties of universal approximation [65] and best approximation [29]. That is, a finite set of RBFs with a single hidden layer can approximate any continuous functions of a finite number of real variables, and one of its approximations to a function has the minimum error between the approximation and the function. An overview of RBFs is given in [3, 47, 27]. We use Gaussian RBFs as the spatial basis functions, which is given by

$$\phi_m(\mathbf{r}) = \exp(-\frac{\|\mathbf{r} - \mathbf{c}_m\|^2}{2\sigma_m^2}), \tag{62}$$

where $\mathbf{c}_m$ is the center and σ_m is the width. Suppose we describe process y using M RBFs such that

$$y(\mathbf{r}) = \sum_{m=1}^{M} \eta_m \phi_m(\mathbf{r}), \tag{63}$$

where η_m are weights and M is the number of RBFs.

The resolution for parametric MT depends on the characteristics of basis functions. In the approximation of a flow field using GRBFs, the number and the distribution of GRBF centers are very important factors that determine the accuracy [72]. In the following section, we choose five GRBFs, four with their centers at the corners of a rectangular domain and one with its center at the center of the domain. Given the GRBF centers, we choose widths of the GRBFs considering all the centroids of the cells as data points for GRBFs by computing the widths as

$$\sigma_m = \frac{2}{P} \sum_{j=1}^{P} \|\mathbf{r}_j - \mathbf{c}_m\|, \tag{64}$$

which is two times the average distance between data points and each center, where M is the number of GRBFs, $\mathbf{r}_j$ the centroids of the cells, P the number of the cell centroids, and $\mathbf{c}_m$ the centers for GRBFs. More strategies to choose the centers and widths of GRBFs are reviewed in [89].

Remark 4. *The basis functions for parametric MT can be initialized after analysis of temporal and spatial characteristics of flow in either historic data or geophysical ocean models. In the absence of this guidance of appropriate data, higher order basis functions (e.g., polynomials) can be used so that a sufficient number of basis functions can sufficiently approximate the temporal and spatial characteristics of an unknown flow field.*

4.4 Simulation and Experimental Results

This section validates parametric MT through both simulations and experiments by using the same data sets used in Section 3.5: 1) simulations using UMSAs under a simulated flow field, 2) indoor experiments using mobile robots that mimic the motion of UMSAs under a simulated flow field, and 3) a field experiment using a glider deployed off the coast of Georgia in September 2013. As the common setting

in the flow field estimation algorithm in Algorithm 3, $\lambda^k = 0.01$, $\forall k$, and five RBFs (four with their centers at the corners of a mapping domain and one with its center at the center of the domain) are used to represent the flow field as discussed in Section 4.3. To determine the convergence for trajectory tracing and flow field estimation, we empirically choose small thresholds, ϵ_γ and $\epsilon_{\mathbf{h}}$ in Algorithms 2 and 3, respectively. Using parameters $\mathbf{h}$ estimated by parametric MT, estimated parametric flow field map $\overline{\mathbf{f}}_{\mathbf{h}}$ is constructed. Then, given the true flow field $\overline{\mathbf{f}}^\star$, estimation error is computed by $\mathbf{e}_{\mathbf{h}} = \overline{\mathbf{f}}^\star - \overline{\mathbf{f}}_{\mathbf{h}}$. The performance of flow field mapping is evaluated by computing the rms of the errors in the x and y components, and while discussing the results from data sets 1 through 3, we show how MT performs when the assumptions on the vehicle speed and the flow field are violated.

4.4.1 Simulations

First, let us discuss results for parametric MT in the ideal case by simulations as compared to results for non-parametric MT discussed in Section 3.5.1. A simulated "true" flow field and vehicle trajectories under the simulated flow are shown in Figures 7a and 8, respectively. Figure 12 displays the estimated flow field map through parametric MT and Table 3 shows convergence conditions for the algorithms and mapping errors. For the parametric flow field, 5 spatial basis functions constructs a 9×9 grid map of the field (i.e., 81 unknowns reduced to 5 unknowns for either x or y component of flow), so the system of equations become overdetermined. Even though the actual vehicle trajectories do not cover the regions at the left upper and right bottom corners of the domain (see Figure 8), parametric MT can estimate the regions. However, for these regions, comparing the magnitude range of the true flow field with that of the parametric flow field, the resulting flow field map also shows low accuracy. To compare with the results for non-parametric MT, the regions where flow is not constructed in the non-parametric flow field are omitted from the

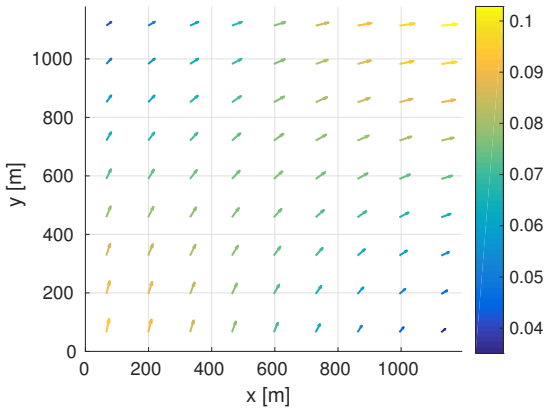


Figure 12: Constructed flow fields from UMSA simulation data.

Table 3: Convergence conditions and quantitative results for parametric MT simulations.

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$
$\epsilon_\gamma = 10^{-2}$	$\epsilon_\mathbf{f} = 30$	$x : 0.011906\, m/s$ $y : 0.0028902\, m/s$

error computation. Considering the mapping errors, although the dependency of MT on the vehicle trajectory is not trivial, MT successfully constructs flow fields from simulated vehicle data with high fidelity in the ideal case. Since parametric MT solves the overdetermined system of equations and basis functions are used to represent a smooth field, we observe that the parametric flow field map has higher accuracy than the non-parametric flow field map.

4.4.2 Indoor experiments

Next, let us discuss results for parametric MT by the indoor experiment using Khepera III mobile robots as compared to results for non-parametric MT discussed in Section 3.5.2. The experimental setup and a simulated "true" field are shown in Figures 9a and 10a. Figure 13 shows a 4×4 grid map of the estimated flow field and Table 4 shows convergence conditions for the algorithms and mapping errors. Compared

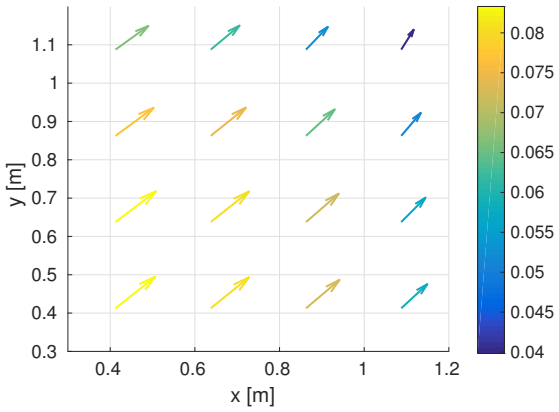


Figure 13: Constructed flow fields from indoor experimental data using Khepera robots under simulated flow.

Table 4: Convergence conditions and quantitative results for parametric MT indoor experiments.

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$
$\epsilon_\gamma = 10^{-4}$	$\epsilon_{\mathbf{f}} = 0.51$	$x : 0.013482\, m/s$ $y : 0.01488\, m/s$

to the magnitude range of the true flow field ([0.03,0.12]), the magnitude range of the parametric flow field ([0.04,0.085]) is smoothed. However, the average of the parametric flow field (0.0484 m/s in x and 0.0462 m/s in y) is very close to that of the true flow field (0.0442 m/s in x and 0.0442 m/s in y). In addition, the mapping errors in Table 4 are small. Compared to the results for non-parametric MT in Section 3.5.2, these results show that under the violation of Assumption 2, the performance difference between non-parametric MT and parametric MT is not significant, but MT can provide the knowledge of flow fields with sufficient accuracy.

4.4.3 A field experiment

Lastly, we discuss parametric MT results using real experimental data collected by an underwater glider in a time-varying flow field as compared to non-parametric MT results in Section 3.5.3. Selected 31 consecutive segments of the glider trajectories are

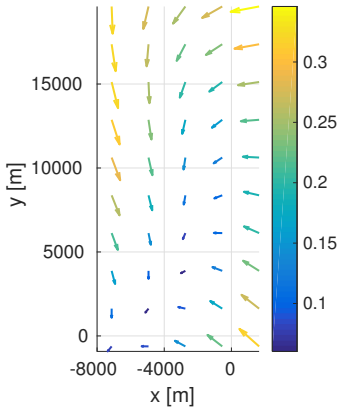


Figure 14: Constructed flow fields from glider experimental data.

shown in Figure 11a and Figure 11b shows the constructed flow field in a 10×5 grid domain using parametric MT after $20 \times K$ iterations. Even though the assumptions are violated, parametric MT is also able to construct a map of a flow field from the experimental data. Solving the overdetermined system of equations and using basis functions, each of which represents the spatial variations of both inner and outer cells, the parametric MT result exhibits an averaged and smoothed field. As is the case with non-parametric MT, the resulting flow field map through parametric MT is also affected by time-varying flow along the vehicle trajectories, requiring MT to incorporate time-varying flow.

CHAPTER V

MOTION TOMOGRAPHY FOR TIME-VARYING FLOW

This chapter presents two approaches for incorporating temporal variability into the MT method. We first resolve the coupling between the temporal and spatial variations of the flow by extending the domain discretization in the MT formulation to both space and time. Another approach assimilates both Eulerian and Lagrangian flow data into a data-driven flow model and estimates parameters of the flow model, unlike classical data assimilation approaches [1, 48] that estimate solutions of PDE-based ocean models. Then, the constructed data-driven flow model provides high-resolution spatio-temporal flow predictions. This chapter presents the methods, challenges, and analysis for both approaches.

5.1 Spatio-Temporal Discretization of a Mapping Domain

5.1.1 Formulation of MT With Spatio-Temporal Discretization

MT can incorporate temporal variability by discretizing the domain in both time and space [10]. Suppose we deploy vehicle i with $\mathbf{r}_i$ denoting its position in domain $\mathcal{D}$ as illustrated in Figure 15. After observation interval $\mathcal{T}$, we obtain the motion-integration error of the vehicle according to Equation (3). Let us discretize $\mathcal{T} = [t^0, t^f]$ into T sub-time intervals such that $\mathcal{T}_1 = [t^0, t^1], \mathcal{T}_2 = [t^1, t^2], \cdots, \mathcal{T}_T = [t^{T-1}, t^f]$, which satisfy $\mathcal{T} = \cup_{\tau=1}^{T} \mathcal{T}_\tau$ and $\cap_{\tau=1}^{T} \mathcal{T}_\tau = \emptyset$. Let us define flow velocity during sub-time interval $\mathcal{T}_\tau$ as $\mathbf{f}_\tau(\mathbf{r}_i) = \mathbf{f}(\mathbf{r}_i, t)$, $t \in \mathcal{T}_\tau$ and assume $\mathbf{f}_\tau(\mathbf{r}_i)$ is constant (e.g., $\mathbf{f}(\mathbf{r}_i, t)$ is evaluated at $t = t_\tau$ for $t \in \mathcal{T}_\tau$) so $\mathbf{f}_\tau(\mathbf{r}_i)$ is piecewise constant in time. Then, for the ith vehicle trajectory γ_i, Equation (3) can be rewritten as

$$\mathbf{d}(\gamma_i, \mathcal{T}) = \sum_{\tau=1}^{T} \int_{\mathcal{T}_\tau} \mathbf{f}_\tau(\mathbf{r}_i) d\tau. \tag{65}$$

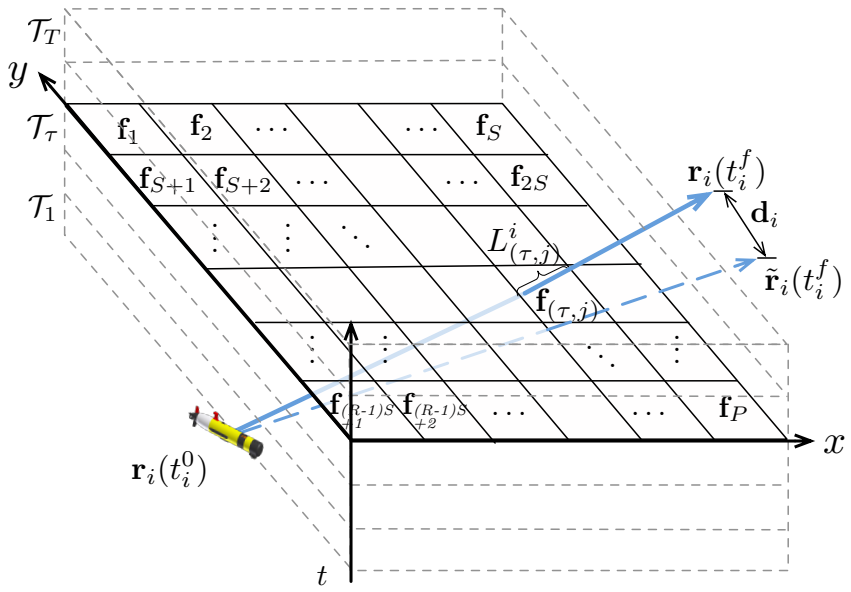


Figure 15: Illustration of MT mapping formulation for a time-varying flow field. Actual (the blue solid line) and predicted (the blue dashed line) vehicle trajectories are displayed in a discretized domain.

We use $\mathbf{d}_i = \mathbf{d}(\gamma_i, \mathcal{T})$ hereafter for simplicity.

Let us discretize the domain $\mathcal{D}$ into $P = R \times S$ grid cells with $D_{(r,s)}$ denoting the (r,s)th cell. We define index $j = (r-1)S + s$ such that $D_j \equiv D_{(r,s)}$, $j = \{1, \cdots, P\}$. We denote the flow velocity in grid cell D_j for sub-time interval $\mathcal{T}_\tau$ by $\mathbf{f}_{(\tau,j)}$ and assume flow within a cell for one sub-time interval is constant. For each trajectory γ_i within grid cell D_j for sub-time interval $\mathcal{T}_\tau$, we introduce arc-length parameter ℓ_i, given by

$$d\ell_i = s^i_{\text{tr}}(\mathbf{f}_{(\tau,j)})dt, \tag{66}$$

in which s^i_{tr} is the speed of the ith vehicle along its actual trajectory. We assume that vehicle heading is constant within a cell for one sub-time interval. Then, the speed of the ith vehicle along the trajectory is given by

$$s^i_{\text{tr}}(\mathbf{f}_{(\tau,j)}) = \left\| s_{\text{h}} \begin{bmatrix} \cos\theta^i_{(\tau,j)} \\ \sin\theta^i_{(\tau,j)} \end{bmatrix} + \mathbf{f}_{(\tau,j)} \right\|, \tag{67}$$

in which $\theta^i_{(\tau,j)}$ is the ith vehicle heading in the jth grid cell for the τth sub-time

interval. Equation (67) leads to the linear trajectory within a cell for one sub-time interval and the piecewise linear trajectory over the domain $\mathcal{D}$.

For the ith vehicle passing through the jth cell for the τth sub-time interval, the length of the vehicle trajectory, $L^i_{(\tau,j)}$, can be obtained by

$$L^i_{(\tau,j)} = \int_{\gamma_i[\mathcal{T}_\tau, D_j]} d\ell_i, \tag{68}$$

in which $\gamma_i[\mathcal{T}_\tau, D_j]$ represents curve γ_i within spatial interval D_j and temporal interval $\mathcal{T}_\tau$. A vehicle trajectory within these spatial and temporal intervals depends on flow in the current cell during the current sub-time interval as well as flow in the previous cells that the trajectory passes through for the previous sub-time intervals. Let us stack all the flow velocities for all the spatio-temporal intervals and define vector $\overline{\mathbf{f}} = [\overline{\mathbf{f}}_x^T, \overline{\mathbf{f}}_y^T]^T$, where $\overline{\mathbf{f}}_x = [\cdots, f_{x,(\tau,1)}, f_{x,(\tau,2)}, \cdots, f_{x,(\tau,P)}, \cdots]^T$ is the x component of all the flow velocities for the spatio-temporal intervals and $\overline{\mathbf{f}}_y = [\cdots, f_{y,(\tau,1)}, f_{y,(\tau,2)}, \cdots, f_{y,(\tau,P)}, \cdots]^T$ is the y component of all the flow velocities for the grid cells. To consider the dependency of the vehicle trajectory on the flow field, we express the length of the vehicle trajectory as $L^i_{(\tau,j)} = L^i_{(\tau,j)}(\overline{\mathbf{f}})$.

By substituting Equation (66) into Equation (65) and following the discretization setting designed above, Equation (65) can be discretized as

$$\mathbf{d}_i = \sum_{\tau=1}^{T} \sum_{j=1}^{P} \frac{L^i_{(\tau,j)}(\overline{\mathbf{f}})}{s^i_{\text{tr}}(\mathbf{f}_{(\tau,j)})} \mathbf{f}_{(\tau,j)}, \quad i = \{1, \cdots, K\}. \tag{69}$$

Considering the flow velocity along the x and y directions separately, we have

$$\begin{aligned} d_{x,i} &= \sum_{\tau=1}^{T} \sum_{j=1}^{P} \frac{L^i_{(\tau,j)}(\overline{\mathbf{f}})}{s^i_{\text{tr}}(\mathbf{f}_{(\tau,j)})} f_{x,(\tau,j)} \\ d_{y,i} &= \sum_{\tau=1}^{T} \sum_{j=1}^{P} \frac{L^i_{(\tau,j)}(\overline{\mathbf{f}})}{s^i_{\text{tr}}(\mathbf{f}_{(\tau,j)})} f_{y,(\tau,j)}. \end{aligned} \tag{70}$$

By constructing vectors $\mathbf{d}_x = [d_{x,1}, d_{x,2}, \cdots, d_{x,K}]^T$ and $\mathbf{d}_y = [d_{y,1}, d_{y,2}, \cdots, d_{y,K}]^T$, we can rewrite Equation (70) as

$$\begin{aligned} \mathbf{d}_x &= \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}}_x \\ \mathbf{d}_y &= \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}}_y, \end{aligned} \tag{71}$$

where $\overline{\mathbf{f}} = [\overline{\mathbf{f}}_x, \overline{\mathbf{f}}_y]$ and

$$\mathbf{L}(\overline{\mathbf{f}}) = \begin{bmatrix} \cdots & \frac{L^1_{(\tau,1)}(\overline{\mathbf{f}})}{s^1_{\text{tr}}(\mathbf{f}_{(\tau,1)})} & \cdots & \frac{L^1_{(\tau,P)}(\overline{\mathbf{f}})}{s^1_{\text{tr}}(\mathbf{f}_{(\tau,P)})} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \cdots & \frac{L^K_{(\tau,1)}(\overline{\mathbf{f}})}{s^K_{\text{tr}}(\mathbf{f}_{(\tau,1)})} & \cdots & \frac{L^K_{(\tau,P)}(\overline{\mathbf{f}})}{s^K_{\text{tr}}(\mathbf{f}_{(\tau,P)})} & \cdots \end{bmatrix}.$$

Equation (71) is nonlinear and underdetermined ($K < T \times P$).

MT creates a map of an underlying time-varying flow field $\mathbf{f}$ by solving Equation (71) through an iterative process that alternates trajectory tracing and flow field estimation [88, 9, 8]. Let us use k to denote the index for iterations. Since vehicle trajectories are typically unknown, starting from an initial guess of estimates for a flow field, trajectory tracing computes vehicle trajectories using the flow estimates and updates matrix $\mathbf{L}$. Then, estimates for a flow field is updated via a row-action approach by iterating the following optimization process with linearized constraints for Equation (71):

$$\begin{aligned} \overline{\mathbf{f}}_x^{k+1} &= \underset{\overline{\mathbf{f}}_x}{\text{argmin}} \frac{1}{2}\|\overline{\mathbf{f}}_x - \overline{\mathbf{f}}_x^k\|^2 \\ \overline{\mathbf{f}}_y^{k+1} &= \underset{\overline{\mathbf{f}}_y}{\text{argmin}} \frac{1}{2}\|\overline{\mathbf{f}}_y - \overline{\mathbf{f}}_y^k\|^2 \end{aligned} \tag{72}$$

subject to

$$\begin{aligned} d_{x,i} &= \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_x \\ d_{y,i} &= \mathbf{L}_i(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_y, \end{aligned} \tag{73}$$

in which $\mathbf{L}_i$ is the i row of matrix $\mathbf{L}$ and $i = \text{mod}(k, K) + 1$.

5.1.2 Parametric MT for Dimension Reduction

Since we obtain Equation (71) by discretizing Equation (3) in both space and time, the dimension of the solution space for $\overline{\mathbf{f}}_x$ and $\overline{\mathbf{f}}_y$ may be very high. In addition, we may not have enough trajectory information to estimate full solution space, causing the solution variable to be spase. To resolve this dimension issue, we incorporate a data-driven flow model in Equation (6). Following the discretization setting for MT,

Equation (6) can be discretized as

$$\begin{aligned} f_{x,(\tau,j)} &= h(\mathbf{r}_j, t_\tau; \Theta_x) = h_{(\tau,j)}(\Theta_x) \\ f_{y,(\tau,j)} &= h(\mathbf{r}_j, t_\tau; \Theta_y) = h_{(\tau,j)}(\Theta_y), \end{aligned} \tag{74}$$

in which $\tau = \{1, \cdots, T\}$ and $j = \{1, \cdots, P\}$ are indices for time and position, respectively, $\mathbf{r}_j$ represents the position for grid cell D_j, and t_τ represents the time for sub-time interval $\mathcal{T}_\tau$. By defining matrix $\mathbf{H}(\cdot)$ such that the element of its (τ, j)th row is $[h_{(\tau,j)}(\cdot)]$, $\overline{\mathbf{f}}_x$ and $\overline{\mathbf{f}}_y$ in Equation (71) can be expressed as

$$\begin{aligned} \overline{\mathbf{f}}_x &= \mathbf{H}(\Theta_x) \\ \overline{\mathbf{f}}_y &= \mathbf{H}(\Theta_y). \end{aligned} \tag{75}$$

By employing a parametric flow model, we have $\mathbf{L}(\overline{\mathbf{f}}) = \mathbf{L}(\Theta)$, in which $\Theta = [\Theta_x^T, \Theta_y^T]^T$. By substituting Equation (75) into Equation (71), the motion-integration error equations become

$$\begin{aligned} \mathbf{d}_x &= \mathbf{L}(\Theta)\mathbf{H}(\Theta_x) \\ \mathbf{d}_y &= \mathbf{L}(\Theta)\mathbf{H}(\Theta_y). \end{aligned} \tag{76}$$

Now, to estimate a flow field, we can solve the equations in Equation (76) for Θ_x and Θ_y instead of solving the equations in Equation (71) for $\overline{\mathbf{f}}_x$ and $\overline{\mathbf{f}}_y$, reducing the dimension of the solution space. In other words, instead of Equation (72), we iteratively solve the following optimization problems:

$$\begin{aligned} \Theta_x^{k+1} &= \underset{\Theta_x}{\operatorname{argmin}} \frac{1}{2}\|\Theta_x - \Theta_x^k\|^2 \\ \Theta_y^{k+1} &= \underset{\Theta_y}{\operatorname{argmin}} \frac{1}{2}\|\Theta_y - \Theta_y^k\|^2 \end{aligned} \tag{77}$$

subject to

$$\begin{aligned} d_{x,i} &= \mathbf{L}_i(\Theta^k)\mathbf{H}(\Theta_x) \\ d_{y,i} &= \mathbf{L}_i(\Theta^k)\mathbf{H}(\Theta_y), \end{aligned} \tag{78}$$

where $i = \operatorname{mod}(k, K) + 1$.

Notice that because of the nonlinearity of $\mathbf{H}(\cdot)$, solving Equation (77) for Θ_x and Θ_y is difficult. As a workaround, we fix spatial or temporal parameters one by one

using the previous estimates and solve Equation (77) for the other parameter. To express the flow model in Equation (5) in a matrix notation, let us define $\Phi(\mathbf{r}) = [\cdots, \phi_m(\mathbf{r}), \cdots]^T$, $m = \{1, \cdots, M\}$ and $\Psi(t) = [\cdots, \psi_n(t), \cdots]^T$, $n = \{1, \cdots, N\}$. Suppose at the kth iteration, we have estimates for parameters, $\rho_{x/y}^k$ and $\eta_{x/y}^k$. By fixing temporal parameters $\rho_{x/y}$ using $\rho_{x/y}^k$, Equation (74) becomes

$$\begin{aligned} f_{x,(\tau,j)} &= (\rho_x^k)^T \Psi(t_\tau)(\Phi(\mathbf{r}_j))^T \eta_x \\ f_{y,(\tau,j)} &= (\rho_y^k)^T \Psi(t_\tau)(\Phi(\mathbf{r}_j))^T \eta_y. \end{aligned} \tag{79}$$

By constructing matrix $\mathbf{H}^{\rho_{x/y}^k}$ such that the element of its (τ, j)th row is $[(\rho_{x/y}^k)^T \Psi(t_\tau)(\Phi(\mathbf{r}_j))^T]$, Equation (75) becomes

$$\begin{aligned} \overline{\mathbf{f}}_x &= \mathbf{H}^{\rho_x^k} \eta_x \\ \overline{\mathbf{f}}_y &= \mathbf{H}^{\rho_y^k} \eta_y. \end{aligned} \tag{80}$$

Let us define $\mathbf{G}^{\rho_{x/y}^k}(\eta) = \mathbf{L}(\eta, \rho^k)\mathbf{H}^{\rho_{x/y}^k}$. Then, we estimate spatial parameters $\eta_{x/y}$ by the following optimization problem:

$$\begin{aligned} \eta_x^{k+1} &= \underset{\eta_x}{\operatorname{argmin}} \frac{1}{2}\|\eta_x - \eta_x^k\|^2 \\ \eta_y^{k+1} &= \underset{\eta_y}{\operatorname{argmin}} \frac{1}{2}\|\eta_y - \eta_y^k\|^2 \end{aligned} \tag{81}$$

subject to

$$\begin{aligned} d_{x,i} &= \mathbf{G}_i^{\rho_x^k}(\eta^k)\eta_x \\ d_{y,i} &= \mathbf{G}_i^{\rho_y^k}(\eta^k)\eta_y. \end{aligned} \tag{82}$$

Following a similar process, by fixing $\eta_{x/y}$, we estimate $\rho_{x/y}$ by the following optimization problem:

$$\begin{aligned} \rho_x^{k+1} &= \underset{\rho_x}{\operatorname{argmin}} \frac{1}{2}\|\rho_x - \rho_x^k\|^2 \\ \rho_y^{k+1} &= \underset{\rho_y}{\operatorname{argmin}} \frac{1}{2}\|\rho_y - \rho_y^k\|^2 \end{aligned} \tag{83}$$

subject to

$$\begin{aligned} d_{x,i} &= \mathbf{G}_i^{\eta_x^k}(\rho^k)\rho_x \\ d_{y,i} &= \mathbf{G}_i^{\eta_y^k}(\rho^k)\rho_y, \end{aligned} \tag{84}$$

where we define $\mathbf{G}^{\eta^k_{x/y}}(\rho) = \mathbf{L}(\eta^k, \rho)\mathbf{H}^{\eta^k_{x/y}}$ and $\mathbf{H}^{\eta^k_{x/y}}$ such that $[(\eta^k_{x/y})^T\Phi(\mathbf{r}_j)(\Psi(t_\tau))^T]$ is the element of its (τ, j)th row.

Based on the above two sub-optimization problems (i.e., Kaczmarz-type methods), we obtain a flow model parameter estimation algorithm (Algorithm 4). In the algorithm, the updating equations include relaxation parameter λ^k which affects the convergence rate. Please note that once we update estimates for either one of the spatial and temporal parameters, we use updated estimates to update estimates for the other parameters. To check the convergence, for k that satisfies $\mathrm{mod}\,(k, K) + 1 = K$, we define residuals $\mathbf{r}^k_{\overline{\mathbf{f}},x} = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k_x - \mathbf{d}_x$ and $\mathbf{r}^k_{\overline{\mathbf{f}},y} = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k_y - \mathbf{d}_y$. We continue updating the solutions until the Euclidean norms of residuals, $\|\mathbf{r}^k_{\overline{\mathbf{f}},x}\|$ and $\|\mathbf{r}^k_{\overline{\mathbf{f}},y}\|$, are sufficiently small (i.e., below a threshold $\epsilon_{\overline{\mathbf{f}}}$). The MT method then create a map of a flow field by iterating a process that alternates flow model parameter estimation and trajectory tracing.

5.1.3 Simulation Results

This section validates the proposed methods through simulations in which we navigate multiple vehicles under flow in a domain of interest. To simulate "true" flow, we use post-processed historic HF-radar data which provide approximately $3 \times 3\ km^2$ spatial resolution and 30-minute temporal resolution. During vehicle navigation, the data are spatially and temporally interpolated along vehicle trajectories. After vehicles travel under this simulated flow, the trajectory information and motion-integration errors of vehicles are used to estimate flow model parameters through the proposed methods.

To parameterize the motion-integration errors in Equation (71), we use a data-driven computational flow model in Equation (6). To initialize basis functions for Equation (6), we select 30-day historical HF-radar data from the time when vehicles start traveling. By decomposing the tidal component of flow from the data using the

Algorithm 4: MT flow model parameter estimation

Data: Motion-integration errors $\mathbf{d}_i, i = \{1, \cdots, K\}$

Set $k = 0$.

Make an initial guess of the solutions, η_x^0, η_y^0, ρ_x^0, and ρ_y^0.

repeat

  for $i = 1$ **to** K **do**

    Update the solutions by

$$\eta_x^{k+1} = \eta_x^k + \lambda^k \frac{d_{x,i} - \mathbf{G}^{\rho_x^k}(\eta^k)\eta_x^k}{\|\mathbf{G}^{\rho_x^k}(\eta^k)\|^2} \left(\mathbf{G}^{\rho_x^k}(\eta^k)\right)^T$$

$$\eta_y^{k+1} = \eta_y^k + \lambda^k \frac{d_{y,i} - \mathbf{G}^{\rho_y^k}(\eta^k)\eta_y^k}{\|\mathbf{G}^{\rho_y^k}(\eta^k)\|^2} \left(\mathbf{G}^{\rho_y^k}(\eta^k)\right)^T$$

$$\rho_x^{k+1} = \rho_x^k + \lambda^k \frac{d_{x,i} - \mathbf{G}^{\eta_x^{k+1}}(\rho^k)\rho_x^k}{\|\mathbf{G}^{\eta_x^{k+1}}(\rho^k)\|^2} \left(\mathbf{G}^{\eta_x^{k+1}}(\rho^k)\right)^T$$

$$\rho_y^{k+1} = \rho_y^k + \lambda^k \frac{d_{y,i} - \mathbf{G}^{\eta_y^{k+1}}(\rho^k)\rho_y^k}{\|\mathbf{G}^{\eta_y^{k+1}}(\rho^k)\|^2} \left(\mathbf{G}^{\eta_y^{k+1}}(\rho^k)\right)^T.$$

    Let $k = k + 1$.

  end

  Construct $\overline{\mathbf{f}}_x$ and $\overline{\mathbf{f}}_y$ using updated estimates for the parameters.

  $\mathbf{r}_{\overline{\mathbf{f}},x}^k = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_x^k - \mathbf{d}_x$.

  $\mathbf{r}_{\overline{\mathbf{f}},y}^k = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}_y^k - \mathbf{d}_y$.

until *a stopping condition is met (e.g.,* $\|\mathbf{r}_{\overline{\mathbf{f}},x}^k\|, \|\mathbf{r}_{\overline{\mathbf{f}},y}^k\| \leq \epsilon_{\overline{\mathbf{f}}}$*)*

T_Tide MATLAB® toolbox [66], we construct temporal basis functions for the tidal component of flow. For the data of the selected period and region, T_Tide extracts total 29 tidal constituents and 3 major constituents ($M2$, $N2$, and $S2$) are used. Considering the tidal residual and pairs of sine and cosine functions, $N = 3{\times}2{+}1$. For the non-tidal component, we use 0th to 2nd order Laguerre polynomials as temporal basis functions (i.e., $L = 3$). To account for spatial variations of flow, $M = 3$ radial basis functions are used.

Remark 5. *Although we use the historic HF-radar data from the time period starting from the beginning time of vehicle navigation, please note that these data are used only for the initialization of basis functions. For parameter estimation, we use the proposed*

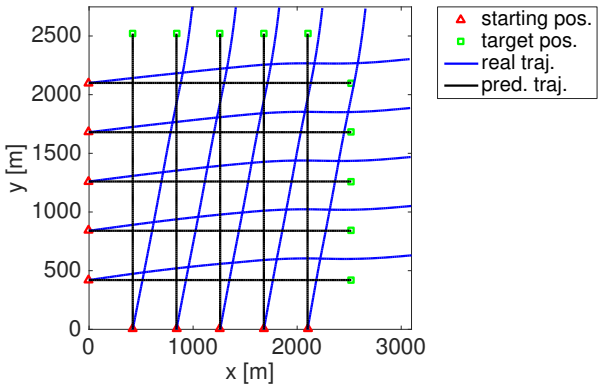


Figure 16: Vehicle trajectories in a domain of interest. The predicted trajectory of a vehicle, connecting its starting position (a red triangle) and target position (a cyan rectangle), is displayed in a black line along with its real trajectory in a blue line.

methods.

After the basis functions for Equation (6) is initialized, we navigate multiple vehicles in a domain. In the simulations, total $K = 10$ vehicles navigate through the domain for 2 hours: 5 from left to right and the other 5 from bottom to top. Figure 16 shows simulated trajectories of the vehicles. Each vehicle travels at 0.35 m/s towards its target position and we assume that the positions of the vehicle are known only at the starting and ending times of its navigation. After vehicles complete their navigation, the trajectory information and motion-integration errors are collected from the vehicles to construct a map of an underlying flow field through MT.

For 10 vehicles traveling for 2 hours, the domain is discretized into $P = 5 \times 5$ grid cells and $T = 5$ sub-time intervals. After the trajectory information of vehicles is collected, flow model parameters are estimated by Algorithm 4 starting from $\eta_x^0 = \eta_y^0 = \rho_x^0 = \rho_y^0 = 0.1$. Figure 17 compares the x and y components of real flow at the center $(x, y) = (1260\,m, 1260\,m)$ of the domain and those of estimated flow constructed using parameters that are computed through MT. The figure illustrates that the estimated flow closely follows the trend of the real flow. Table 5 shows

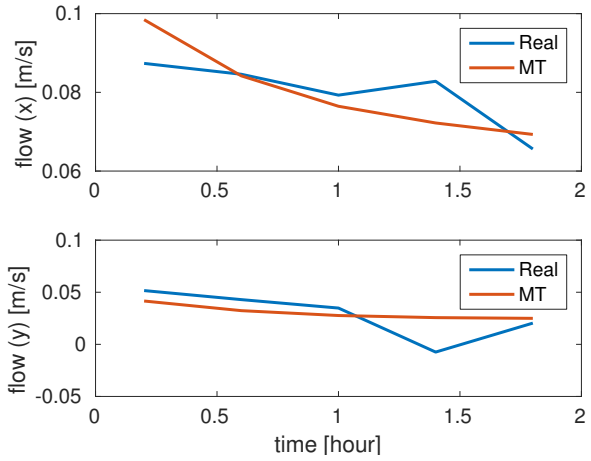


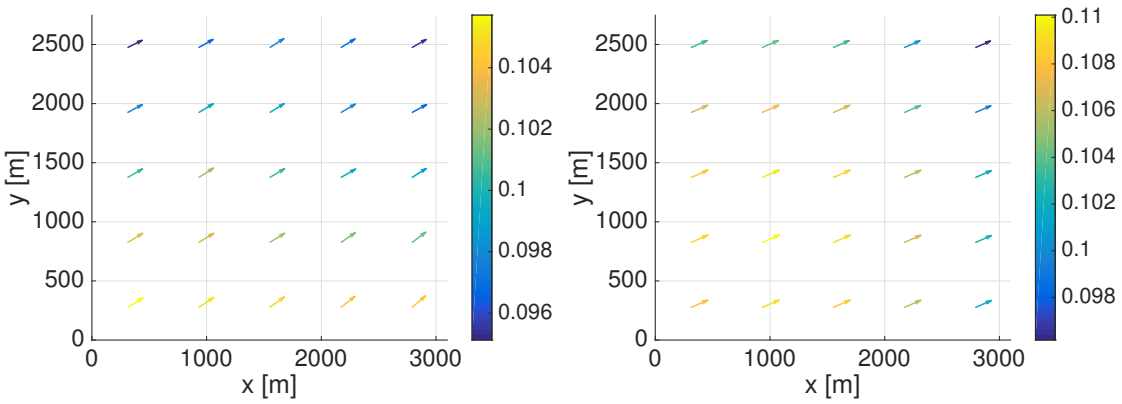
Figure 17: Comparison between real flow and estimated flow constructed using parameters computed through non-distributed MT at the center of the domain where vehicles navigate.

Table 5: Convergence conditions and quantitative results for parametric MT simulations with time-varying flow at the center of the domain

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$
$\epsilon_\gamma = 10^{-2}$	$\epsilon_{\overline{\mathbf{f}}} = 25$	$x : 0.0072\, m/s$ $y : 0.0166\, m/s$

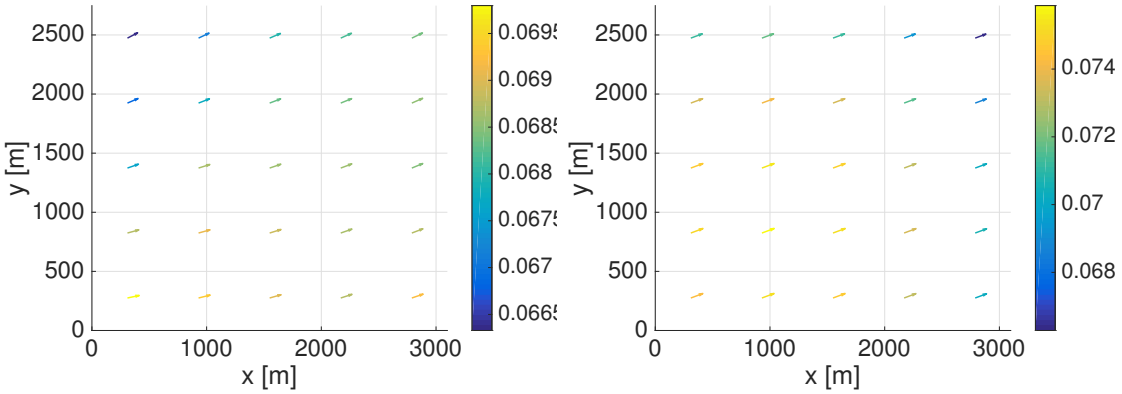
convergence conditions for the algorithms and mapping errors. The rms error for the x component of the estimated flow is 0.0072 m/s and that for the y component of the estimated flow is 0.0166 m/s. To reduce the error, we can increase the number of spatial and temporal basis functions, but we used a minimal number of basis functions ($N = 7$, $L = 3$, and $M = 3$) to balance the dimension of the parameters and the flow mapping accuracy.

Figure 18 compares "true" and estimated flow fields at 0.2 hour and at 1.8 hour, respectively. The range of the colorbar represents flow velocities. By comparing the range of the colorbar at two different times, we can see that the flow field is time-varying. Table 6 shows convergence conditions for the algorithms and mapping errors at two different times, $t = 0.2$ hour and $t = 1.8$ hour. The rms errors between



(a) True field at $t = 0.2$ hour.

(b) Estimated field at $t = 0.2$ hour.



(c) True field at $t = 1.8$ hour.

(d) Estimated field at $t = 1.8$ hour.

Figure 18: True and estimated flow fields at $t = 0.2$ hour and $t = 1.8$ hour.

Table 6: Convergence conditions and quantitative results for parametric MT simulations with time-varying flow at time $t = \{0.2, 1.8\}$ hour

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$ ($t = 0.2$ hour)	$\mathbf{e}^{rms}$ ($t = 1.8$ hour)
$\epsilon_\gamma = 10^{-2}$	$\epsilon_{\overline{\mathbf{f}}} = 25$	$x : 0.0118\, m/s$ $y : 0.0123\, m/s$	$x : 0.0045\, m/s$ $y : 0.0050\, m/s$

"true" and estimated flow fields at 0.2 hour are 0.0118 m/s in the x direction and 0.0123 m/s in the y direction. The rms errors at 1.8 hour are 0.0045 m/s in the x direction and 0.0050 m/s in the y direction. These results show that the MT method can successfully incorporate temporal variability by discretizing the domain in both space and time.

5.2 Assimilation of Eulerian Data Along With Lagrangian Data From UMSAs

To incorporate temporal variability into a flow field map estimated by MT, we introduce generic environmental models (GEMs), which are data-driven computational models combined with real-time data streams collected from UMSAs [11]. The data-driven flow models developed in the author's previous work [7, 12] including MT [88, 9, 8] fall into GEMs. We assimilate both Eulerian data collected from stationary sensors and Lagrangian data collected from UMSAs into a GEM.

As a general framework for a GEM, we use a data-driven flow model in Equation (6) and define a state variable Θ to be either Θ_x or Θ_y. Suppose a mooring provides Eulerian flow data with sampling period T_s to us. In addition, suppose a UMSA navigates near the mooring, providing Lagrangian flow data that are converted to a spatial flow map through MT every αT_s, $\alpha \in \mathbb{N}^+ \geq 2$ where $\mathbb{N}^+$ denotes positive integer. Since the time scale of mooring data is smaller, we let $\Theta_k = \Theta(kT_s)$ where k is the time step index. To account for the inaccuracy of the ocean modeling techniques such as approximation error, we consider

$$\Theta_{k+1} = \Theta_k + w_k, \tag{85}$$

where w_k is the process noise at the kth step. We assume w_k is Gaussian with zero mean and known covariance Q_k.

For time step k and position $\mathbf{r}$, let us define an observation variable $z_k(\mathbf{r})$ to be observation of either $f_x(\mathbf{r}, kT_s)$ or $f_y(\mathbf{r}, kT_s)$. Provided that a mooring is deployed at $\mathbf{r}^E$, Eulerian flow observation at time step k is denoted by $z_k^E(\mathbf{r}^E)$. We use MT to convert the Lagrangian data collected from a UMSA to a spatial map of the flow field. Suppose the map of the flow field contains P grid cells (see Figure 6). Let $\mathbf{r}_j^L$ be a position that represents the jth grid cell. Then, we denote flow observation based on the Lagrangian data at time step k by $z_k^L(\mathbf{r}_j^L)$, $j = \{1, \cdots, P\}$.

Let us stack both Eulerian and Lagrangian observations such that $z_k = [z_k^E(\mathbf{r}^E), z_k^L(\mathbf{r}_1^L), \cdots, z_k^L(\mathbf{r}_P^L)]$ and define matrix H such that

$$H_k(\Theta_k) = \begin{bmatrix} h_k(\mathbf{r}^E, \Theta_k) \\ h_k(\mathbf{r}_1^L, \Theta_k) \\ \vdots \\ h_k(\mathbf{r}_P^L, \Theta_k) \end{bmatrix}, \tag{86}$$

where $h_k(\mathbf{r}, \Theta) = h(\mathbf{r}, kT_s; \Theta)$. Then, the observation equation at time step k is given by

$$z_k = H_k(\Theta_k) + v_k, \tag{87}$$

where the observation noise v_k is assumed Gaussian with zero mean and known covariance matrix R_k.

Even though state equation in Equation (85) is linear, we have nonlinear observation equation in Equation (87). Therefore, estimation of parameters becomes a nonlinear filtering problem. To solve this nonlinear filtering problem, we separate spatial parameter estimation and temporal parameter estimation. In other words, we decompose a nonlinear filtering problem into two linear sub-filtering problems. Then, we consider one set of parameters as constant and solve the filtering problem for the other set of parameters. Since Lagrangian data through MT can provide higher spatial variations than Eulerian data from a mooring, we use Lagrangian data to update spatial parameters and Eulerian data to update temporal parameters.

5.2.1 Spatial Parameter Estimation

Suppose the basis functions for flow model in Equation (6) and the previous estimates of parameters are known. Let us omit subscripts x and y for simplicity of presentation. Given an estimated map of the flow field computed through MT from Lagrangian data, we fix all the temporal parameters ρ using the previous estimates, $\hat{\rho}$, and estimate spatial parameters η. Then, in state vector Θ, η is the only unknown. Let

us define $\Phi(\mathbf{r}) = [\cdots, \phi_m(\mathbf{r}), \cdots]^T$, $m = \{1, \cdots, M\}$ and $\Psi(t) = [\cdots, \psi_n(t), \cdots]^T$, $n = \{1, \cdots, N\}$. Then, flow velocity can be expressed as

$$f(\mathbf{r}, t) = \hat{\rho}^T \Psi(t) \Phi(\mathbf{r})^T \eta \tag{88}$$

which are linear in η. That is, the filtering problem for η is now linear. Suppose previous estimates for ρ are computed at time step $k-1$. Then, the state and observation equations for spatial parameter estimation are given by

$$\eta_{k+1} = \eta_k + w_k^\eta \tag{89}$$

$$z_k = H_k^\eta \eta_k + \nu_k^\eta, \tag{90}$$

where w_k^η is the process noise for η at the kth step, which is Gaussian with zero mean and known covariance Q_k^η and

$$H_k^\eta = \begin{bmatrix} \rho_{k-1}^T \Psi_k \Phi(\mathbf{r}_1^L)^T \\ \vdots \\ \rho_{k-1}^T \Psi_k \Phi(\mathbf{r}_P^L)^T \end{bmatrix}. \tag{91}$$

Then, the filtering equations for spatial parameters are given by

$$\begin{aligned}
\hat{\eta}_k^- &= \hat{\eta}_{k-1}^+ \\
P_k^{\eta-} &= P_{k-1}^{\eta+} + Q_{k-1}^\eta \\
K_k &= P_k^{\eta-} (H_k^\eta)^T (H_k^\eta P_k^{\eta-} (H_k^\eta)^T + R_k^\eta)^{-1} \\
\hat{\eta}_k^+ &= \hat{\eta}_k^- + K_k (z_k - H_k^\eta \hat{\eta}_k^-) \\
P_k^{\eta+} &= (I - K_k H_k^\eta) P_k^{\eta-} (I - K_k H_k^\eta)^T + K_k R_k^\eta K_k^T,
\end{aligned}$$

where $\hat{\eta}_k$ is the optimal estimate of η_k.

5.2.2 Temporal Parameter Estimation

We use Eulerian data from a mooring to update temporal parameters. In contrast to the spatial parameter estimation, when Eulerian data are available, we fix all the

spatial parameters η using the previous estimates, $\hat{\eta}$. Suppose Eulerian data are available at time step k and previous estimates for spatial parameters were computed at time step $\lfloor k/\alpha \rfloor \alpha$. For temporal parameter estimation, the state and observation equations are given by

$$\rho_{k+1} = \rho_k + w_k^\rho \tag{92}$$

$$z_k = H_k^\rho \rho_k + \nu_k^\rho, \tag{93}$$

where w_k^ρ is the process noise for ρ at the kth step, which is Gaussian with zero mean and known covariance Q_k^ρ and

$$H_k^\rho = \eta_{\lfloor k/\alpha \rfloor \alpha}^T \Phi(\mathbf{r}^E) \Psi_k^T. \tag{94}$$

For temporal parameter estimation, the filtering equations are given by

$$\begin{aligned}
\hat{\rho}_k^- &= \hat{\rho}_{k-1}^+ \\
P_k^{\rho-} &= P_{k-1}^{\rho+} + Q_{k-1}^\rho \\
K_k &= P_k^{\rho-} (H_k^\rho)^T (H_k^\rho P_k^{\rho-} (H_k^\rho)^T + R_k^\rho)^{-1} \\
\hat{\rho}_k^+ &= \hat{\rho}_k^- + K_k (z_k - H_k^\rho \hat{\rho}_k^-) \\
P_k^{\rho+} &= (I - K_k H_k^\rho) P_k^{\rho-} (I - K_k H_k^\rho)^T + K_k R_k^\rho K_k^T,
\end{aligned}$$

where $\hat{\rho}_k$ is the optimal estimate of ρ_k.

Eulerian data and Lagrangian data may have different time scales, so we cannot always update both spatial and temporal parameters at the same time. Therefore, although spatial parameter estimates may not be up-to-date, we use the latest available spatial parameter estimates to evaluate the spatial parameters.

5.2.3 Observability Analysis

Given a system with no control input, a Kalman filter converges if the system is uniformly completely observable [40]. We redefine uniform complete observability in [40].

Definition 1. *The linear system*

$$\Theta_{k+1} = \Theta_k + w_k \tag{95}$$

$$z_k = H_k\Theta_k + \nu_k \tag{96}$$

is uniformly completely observable if there exist $\tau, \beta_1, \beta_2 > 0$ *such that the observability Gramian* $\mathcal{J}(k, k-\tau) = \sum_{j=k-\tau}^{k} H_j^T R_j^{-1} H_j$ *satisfies* $\beta_1 I \preccurlyeq \mathcal{J}(k, k-\tau) \preccurlyeq \beta_2 I$ *for all* $k > \tau$ *where the dimension of* I *is defined accordingly. Here,* R_j *is the covariance matrix for noise* ν_j*.*

Lemma 3. *Let nonzero vectors* $\mathbf{u}_i \in \mathbb{R}^n$*,* $i = \{1, \cdots, n\}$ *be linearly independent. Then,* $\mathcal{M} = \sum_{i=1}^{n} \mathbf{u}_i \mathbf{u}_i^T \in \mathbb{R}^{n\times n}$ *has full rank.*

Proof. Consider nonzero vector $\mathbf{v}_1 \in \text{span}\{\mathbf{u}_2, \cdots, \mathbf{u}_n\}^{\perp}$. Then, $\mathcal{M}\mathbf{v}_1 = (\mathbf{u}_1\mathbf{u}_1^T)\mathbf{v}_1 = \mathbf{u}_1(\mathbf{u}_1^T\mathbf{v}_1)$ is a nonzero scalar multiple of $\mathbf{u}_1$. Similarly, for $\mathbf{v}_i \in \text{span}\{\mathbf{u}_j\}_{j\neq i}^{\perp}$, $\mathcal{M}\mathbf{v}_i$ is a nonzero scalar multiple of $\mathbf{u}_i$. In other words, nonzero scalar multiples of each $\mathbf{u}_i$ are in the range of $\mathcal{M}$ and the dimension of the range of $\mathcal{M}$ is n, which is equivalent to $\mathcal{M}$ having full rank (c.f., [37], pg. 13). □

In the following theorem, we prove uniform complete observability for spatial parameter estimation:

Theorem 4. *The system in Equations* (89) *and* (90) *is uniformly completely observable if the following conditions are satisfied:*

(Cd1) *The matrix* R_j^{η} *is uniformly bounded for all* j *(i.e.,* $\beta_3 I_{(P+1)\times(P+1)} \preccurlyeq R_j^{\eta} \preccurlyeq \beta_4 I_{(P+1)\times(P+1)}$ *for some constants* $\beta_3, \beta_4 > 0$*.*

(Cd2) *Among* $\Phi(\cdot)$*'s evaluated at the position of the mooring,* $\mathbf{r}^E$*, and the positions of the grid cells,* $\mathbf{r}_i^L$*,* $i = \{1, \cdots, P\}$*, at least* M $\Phi(\cdot)$*'s are linearly independent.*

Remark 6. *Condition* (Cd2) *can be satisfied by choosing spatial basis functions appropriately. Consider Gaussian radial basis functions (GRBFs) indexed by* m*,*

$\phi_m(\mathbf{r}) = \exp\left(-\frac{\|\mathbf{r}-\mathbf{c}_m\|}{2\sigma^2}\right)$, *where* $\mathbf{c}_m$ *is the center and* σ *is the width. If we use GRBFs as spatial basis functions for* $\Phi(\mathbf{r}) = [\cdots, \phi_m(\mathbf{r}), \cdots]^T$, $m = \{1, \cdots, M\}$, (Cd2) *can be satisfied by choosing* M *different centers.*

Proof. For the system in Equations (89) and (90), we have observability Gramian $\mathcal{J}_\eta(k, k-\tau) = \sum_{j=k-\tau}^{k} (H_j^\eta)^T (R_j^\eta)^{-1} H_j^\eta$. From (Cd1), R_j^η is positive definite and bounded above and below. Therefore, we obtain $\beta_4^{-1} \sum_{j=k-\tau}^{k} (H_j^\eta)^T H_j^\eta \preccurlyeq \mathcal{J}_\eta(k, k-\tau) \preccurlyeq \beta_3^{-1} \sum_{j=k-\tau}^{k} (H_j^\eta)^T H_j^\eta$ for all $k > \tau$. Next, we prove that $\sum_{j=k-\tau}^{k} (H_j^\eta)^T H_j^\eta$ is positive definite. We can compute

$$
\begin{aligned}
(H_j^\eta)^T H_j^\eta &= \begin{bmatrix} \rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_1^L)^T \\ \vdots \\ \rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_P^L)^T \end{bmatrix}^T \begin{bmatrix} \rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_1^L)^T \\ \vdots \\ \rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_P^L)^T \end{bmatrix} \\
&= \left(\rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_1^L)^T\right)^T \rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_1^L)^T + \cdots + \left(\rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_P^L)^T\right)^T \rho_{j-1}^T \Psi_j \Phi(\mathbf{r}_P^L)^T \\
&= \left(\rho_{j-1}^T \Psi_j\right)^2 \left[\Phi(\mathbf{r}_1^L)\Phi(\mathbf{r}_1^L)^T + \cdots + \Phi(\mathbf{r}_P^L)\Phi(\mathbf{r}_P^L)^T\right].
\end{aligned}
$$

The quadratic scalar term $\left(\rho_{j-1}^T \Psi_j\right)^2$ is always positive unless the temporal component of flow is zero everywhere. Let us define $\mathcal{M}^\eta = [\Phi(\mathbf{r}_1^L)\Phi(\mathbf{r}_1^L)^T + \cdots + \Phi(\mathbf{r}_P^L)\Phi(\mathbf{r}_P^L)^T]$. All the terms inside the bracket are rank-one positive-semidefinite matrices. From (Cd2) and Lemma 3, $\mathcal{M}^\eta$ has full rank and is positive definite. Therefore, $\beta_5 I_{M\times M} \preccurlyeq \mathcal{M}^\eta \preccurlyeq \beta_6 I_{M\times M}$ for some constants $\beta_5, \beta_6 > 0$. □

Remark 7. *Suppose we use multiple UMSAs to collect Lagrangian data. Regardless of the number of UMSAs, the number of grid cells does not change, preserving the dimension of observation matrix* H^η*. However, using multiple UMSAs will increase the rate of visiting all the grid cells and the chance of generating linearly independent* $\Phi(\cdot)$*'s. Therefore, the uniform complete observability condition can be satisfied in shorter time window than using just one UMSA.*

In the following theorem, we prove uniform complete observability for temporal parameter estimation:

Theorem 5. *The system in Equations* (92) *and* (93) *is uniformly completely observable if* (Cd1) *and the following condition are satisfied:*
(Cd3) *The sampling period T_s is chosen such that at least N Ψ_j's, $j \in [k-\tau, k] \subset \mathbb{N}^+$ are linearly independent.*

Proof. For the system in Equations (92) and (93), we have observability Gramian $\mathcal{J}_\rho(k, k-\tau) = \sum_{j=k-\tau}^{k} (H_j^\rho)^T (R_j^\rho)^{-1} H_j^\rho$. From (Cd1), R_j^ρ is positive definite and bounded above and below. Therefore, we obtain $\beta_4^{-1} \sum_{j=k-\tau}^{k} (H_j^\rho)^T H_j^\rho \preccurlyeq \mathcal{J}_\rho(k, k-\tau) \preccurlyeq \beta_3^{-1} \sum_{j=k-\tau}^{k} (H_j^\rho)^T H_j^\rho$ for all $k > \tau$. Next, we prove that $\sum_{j=k-\tau}^{k} (H_j^\rho)^T H_j^\rho$ is positive definite. We can compute

$$\begin{aligned}(H_j^\rho)^T H_j^\rho &= \left[\begin{array}{c} \eta_{\lfloor j/\alpha \rfloor \alpha}^T \Phi(\mathbf{r}^E) \Psi_j^T \end{array} \right]^T \left[\begin{array}{c} \eta_{\lfloor j/\alpha \rfloor \alpha}^T \Phi(\mathbf{r}^E) \Psi_j^T \end{array} \right] \\ &= \left(\Phi(\mathbf{r}^E)^T \eta_{\lfloor j/\alpha \rfloor \alpha} \eta_{\lfloor j/\alpha \rfloor \alpha}^T \Phi(\mathbf{r}^E) \right) \Psi_j \Psi_j^T\end{aligned}$$

The quadratic term in the parenthesis is always positive unless the spatial component of flow is zero everywhere. Let $\mathcal{M}^\rho = \sum_{j=k-\tau}^{k} \Psi_j^T \Psi_j$. From (Cd3) and Lemma 3, $\mathcal{M}^\rho$ has full rank and is positive definite. Therefore, $\beta_7 I_{N \times N} \preccurlyeq \mathcal{M}^\rho \preccurlyeq \beta_8 I_{N \times N}$ for some constants $\beta_7, \beta_8 > 0$. □

Remark 8. *Eulerian data from multiple moorings can be assimilated into a data-driven flow model. The increased number of moorings will increase the dimension of the observation matrix H^ρ for Eulerian data. We can choose locations of moorings such that the rank of the observation matrix increases. The uniform complete observability condition for Eulerian data assimilation using multiple moorings can be satisfied over a shorter time window than using just one mooring.*

Remark 9. *Even though the system in Equations* (89) *and* (90) *and the system in Equations* (92) *and* (93) *are uniformly completely observable, the convergence rate and accuracy may depend on how observable the systems are. For this problem, we can incorporate observability metrics such as the unobservability index [45].*

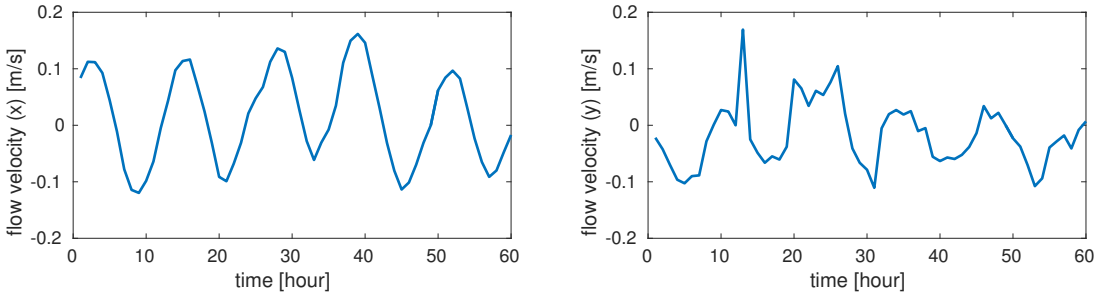
5.2.4 Simulation Results

This section validates the proposed data assimilation method through simulations. To show that the proposed method resolves temporal variations for MT by incorporating both Eulerian and Lagrangian data, we compare two implementations of GEMs: GEM-EL assimilating both Eulerian and Lagrangian data and GEM-L assimilating Lagrangian data only. Both methods process Lagrangian data through MT.

For a data-driven flow model that assimilates Eulerian data and Lagrangian data, we use the flow model in Equation (5). For spatial basis functions, we use Gaussian radial basis functions (RBFs). For temporal basis functions, we first decompose ocean flow into tidal and non-tidal components. Since the tidal component of ocean flow can be described by a superposition of multiple tidal constituents characterized by magnitude and phase, we use a series of sinusoidal basis functions as temporal basis functions for tidal flow. For non-tidal flow, weighted Laguerre polynomials are chosen as temporal basis functions.

To initialize the flow model, we use post-processed historic HF-radar data. The data provide 3×3 km^2 spatial resolution and 30-minute temporal resolution. In our simulation, one hour corresponds to one time step for parameter estimation. We decompose the tidal component of flow from HF-radar sea surface currents using the T_Tide MATLAB® toolbox [66]. Then, 30-day historical data are used to initialize the flow model. For the data of the selected period and region, T_Tide extracts total 29 tidal constituents and $N = 3$ major constituents ($M2$, $N2$, and $S2$) are used to generate temporal basis functions for the tidal component in the model. For the non-tidal component, we use 0- to $L = 4$th order Laguerre polynomials as temporal basis functions. To account for spatial variations of flow, $M = 5$ RBFs are used.

After the model is initialized, we select another segment of HF-radar data next to the period of the initialization data set to simulate the "true" flow fields. A mooring is deployed at a fixed location $(0, 0)$ in a domain of the simulated fields. Figure 19



(a) x component of flow.

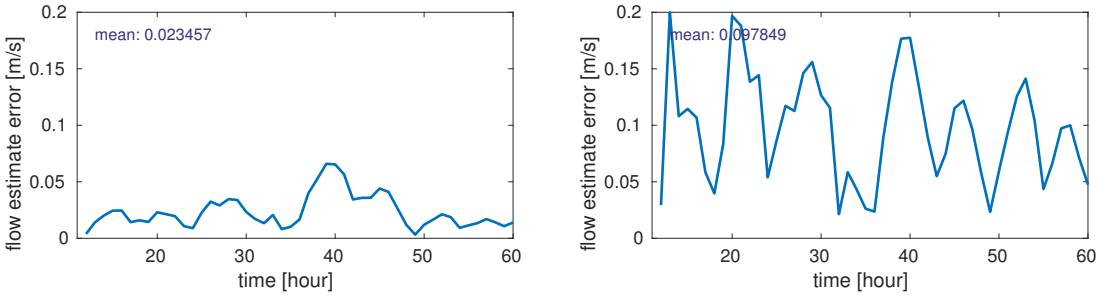
(b) y component of flow.

Figure 19: Time-series Eulerian flow data collected from a mooring.

shows flow velocities at the location of the mooring for 60 time steps. We assume mooring data are available each time step. Around the mooring, a UMSA always travels across the mooring (i.e., its desired heading is towards the mooring) over one observation interval which is one time step, criss-crossing the domain. However, to more thoroughly address spatial variations of flow, vehicle trajectories are collected over multiple time steps prior to the MT mapping.

For estimation of the flow model parameters, we randomly generate initial spatial and temporal parameters on $[-0.5, 0.5]$. To update spatial parameters using Lagrangian data processed through the MT mapping, we first discretize the MT mapping domain into $P = 6 \times 6$ grid cells. Since the most dominant tidal constituent for the simulated flow fields is $M2$, the principal lunar component with a period of 12.42 hours, to reduce the influence of temporal variations of flow from the mapping results, we create a map of a flow field every $\alpha = 12$ time steps. Then, after fixing temporal parameters using previous estimates, P observations are assimilated to estimate spatial parameters. For temporal parameter estimation, we fix spatial parameters with previous estimates and assimilate Eulerian data from the mooring every time step to update the estimates of temporal parameters. To apply the proposed sub-filters, we let $Q_k^\rho = 10^{-4} I_{(L+2N+2)\times(L+2N+2)}$, $Q_k^\eta = 10^{-4} I_{M\times M}$, and $R_k = 10^{-4} I_{P+1}$.

To validate the proposed method, we compute the error between the simulated



(a) GEM-EL. (b) GEM-L.

Figure 20: Spatial mean square error for GEM-EL and GEM-L.

"true" flow fields and the estimated flow fields. Then, spatial mean of the square root of the errors are computed. Figure 20 shows the errors for GEM-EL and GEM-L from 12 to 60 time steps. Since Lagrangian data are assimilated after the first 12 time steps, we display errors from time step $k = 12$. Considering the period of $M2$ tidal constituents, to emphasize the importance of assimilating Eulerian data from moorings for temporal variations, we compute the error every 6 time steps. Since GEM-EL incorporates both spatial and temporal variations of flow, overall it shows lower values and smaller variance of the errors than GEM-L. Temporally averaged errors are displayed in the figures. Over the period of 12 to 60 time steps, the proposed method shows 76 percent lower error. Based on these results, we validate that the method actually resolves temporal variations for MT.

To demonstrate the effectiveness of resolving temporal variations for MT, we simulate two virtual mooring vehicles: one guided by GEM-EL and the other guided by GEM-L. Both vehicles are first deployed at $(0, 0)$ where the mooring is installed in the domain. Then, after the first assimilation of Lagrangian data (i.e., after time step $k = 12$), they start attempting to cancel the flow using estimated flow using GEM-EL and GEM-L until time step $k = 60$. Here, we assume that perfect flow cancellation by a UMSA is possible regardless of the strength of flow. Figure 21 shows the trajectories of both vehicles. By incorporating both temporal and spatial variations of flow,

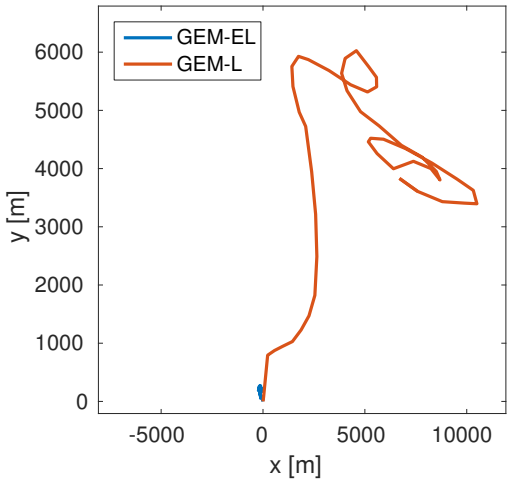


Figure 21: Virtual mooring vehicles guided by GEM-EL and GEM-L.

the vehicle guided by GEM-EL successfully stays within a region close to the origin. However, with no knowledge of spatial variation of flow, the vehicle guided by GEM-L fails to stay close to the origin. These results show that incorporating Eulerian data resolves temporal variations for MT and the resulting data-driven flow model that assimilates both Eulerian and Lagrangian data can improve the navigation accuracy for UMSAs.

CHAPTER VI

DISTRIBUTED MOTION TOMOGRAPHY

Let us refer to vehicles that form a collaborative data network as nodes. Solving Equation (20) for $\overline{\mathbf{f}}$ and solving Equation (51) for $\mathbf{h}$ require complete knowledge of $\mathbf{d}$ and $\mathbf{L}$ collected from all the nodes. However, because of the limited communication capabilities of the nodes or the large scale of applications, data may be collected or shared among nodes in a distributed manner. Suppose the ith node has partial knowledge of $\mathbf{d}$ and $\mathbf{L}$, denoted by d_i and $\mathbf{L}_i$. We can view Equations (32) and (53) that correspond to solving Equation (20) for $\overline{\mathbf{f}}$ and solving Equation (51) for $\mathbf{h}$, respectively, as joint optimization problems where the ith node solves the ith problem for an estimate of $\overline{\mathbf{f}}$ or $\mathbf{h}$ shared among all the nodes. Suppose the estimated solution is not shared and the ith node solves the ith problem for its own estimate of $\overline{\mathbf{f}}$ or $\mathbf{h}$ based on d_i and $\mathbf{L}_i$ only. Then, the solution is sub-optimal and inevitably biased. To resolve this issue, we develop a distributed method for solving Equations (20) and (51) in this section.

A course of distributed methods for linear least squares problems are well summarized in [76, 71]. For distributed implementation of the Kaczmarz method, Kamath et al. [44] proposed a parallel algorithm for a randomized Kaczmarz method [81] that requires a data aggregation center and extended it to a distributed algorithm in which data are shared only with a set of neighbors of the nodes in a network. The convergence for both algorithms is proved and the distributed method is tested through seismic tomography. However, the problem considered therein is a linear system of equations, so their method is not directly applicable to MT.

For a nonlinear system of equations, the relevant literature [62, 82, 55, 14, 22, 39]

typically focuses on unconstrained (or set-constrained) optimization problems whose global objective functions that are mostly assumed convex are denoted by $\mathfrak{f}(\mathbf{x})$ in which $\mathbf{x} \in \mathbb{R}^d$ and $\mathfrak{f} : \mathbb{R}^d \to \mathbb{R}$. A common framework of the distributed approach for these problems in the references is that the objective function is separable such that $\mathfrak{f}(\mathbf{x})$ is the summation of local cost functions $\mathfrak{f}_i(\mathbf{x})$ only known to their corresponding nodes (i.e., $\mathfrak{f}(\mathbf{x}) \equiv \sum_{i=1}^{N} \mathfrak{f}_i(\mathbf{x})$ where N is the number of nodes in a network). Following this framework, given an equation for MT, $\mathbf{d} = \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}}$, we can formulate objective functions such that $\mathfrak{f}(\overline{\mathbf{f}}) = \|\mathbf{d} - \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}}\|^2$ and $\mathfrak{f}_i(\overline{\mathbf{f}}) = \|d_i - \mathbf{L}_i(\overline{\mathbf{f}})\overline{\mathbf{f}}\|^2$. This can be viewed as a nonlinear least squares problem, which may be more suitable for overdetermined systems. In addition, this formulation is different from the formulation of Kaczmarz-type methods for MT. In this section, considering that the system of equations for MT is underdetermined, we first derive a distributed Kaczmarz method using a framework constructed in this document. Then, this distributed Kaczmarz method is extended to solving a nonlinear system of equations constructed for MT.

6.1 Graph Representation of A Network

We consider a set of K nodes denoted by $\mathcal{V} = \{v_1, \cdots, v_K\}$, each has communication and computation capabilities. The network of nodes is represented by an undirected graph where a communication link between two nodes is referred to as an edge. We denote the graph by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ comprising a set of nodes, $\mathcal{V}$, with a set of edges, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The edge between an unordered pair of nodes v_i and v_j is denoted by $e_{(i,j)} = e_{(j,i)} \in \mathcal{E}$. We assume a sparse and connected graph and define a group of nodes who can communicate with node v_i, including itself, as a set of its neighbors, $\mathcal{N}_i = \{v_i\} \cup \{v_j \in \mathcal{V} | e_{(i,j)} \in \mathcal{E}\}$. Data collected by node v_i are synchronously shared with neighbors in $\mathcal{N}_i$. We assume that the communication links between node v_i and its neighbors are good enough for data exchange (e.g., no data loss).

6.2 A Distributed Kaczmarz Method

Let us consider a linear system of equations

$$\mathbf{d} = \mathbf{L}\overline{\mathbf{f}}, \tag{97}$$

where $\mathbf{d} \in \mathbb{R}^K$ is observations, $\overline{\mathbf{f}} \in \mathbb{R}^P$ is unknowns, and $\mathbf{L} \in \mathbb{R}^{K \times P}$ is an observation matrix. Typically, Equation (97) is an underdetermined system ($K < P$). The Kaczmarz method solves Equation (97) for $\overline{\mathbf{f}}$ by iteratively solving

$$\begin{aligned} \overline{\mathbf{f}}^{k+1} &= \underset{\overline{\mathbf{f}}}{\operatorname{argmin}} \frac{1}{2} \left\| \overline{\mathbf{f}} - \overline{\mathbf{f}}^k \right\|^2 \\ \text{subject to} \quad & d_i = \mathbf{L}_i \overline{\mathbf{f}}, \end{aligned} \tag{98}$$

where $\mathbf{L}_i$ is the ith row of $\mathbf{L}$, d_i is the ith element of $\mathbf{d}$, $i = \operatorname{mod}(k, K) + 1$, and $k = \{0, 1, 2, \cdots\}$ is the index for iterations or communication rounds. This optimization problem leads to

$$\overline{\mathbf{f}}^{k+1} = \overline{\mathbf{f}}^k + \frac{d_i - \mathbf{L}_i \overline{\mathbf{f}}^k}{\|\mathbf{L}_i\|^2} \mathbf{L}_i^T. \tag{99}$$

The convergence of $\overline{\mathbf{f}}^k$ updated by Equation (99) is proved in [15].

To develop a distributed version of the Kaczmarz method, we consider that node v_i solves

$$d_i = \mathbf{L}_i \overline{\mathbf{f}}_i, \tag{100}$$

where $\mathbf{L}_i$ is the ith row of matrix $\mathbf{L}$ and d_i the ith element of vector $\mathbf{d}$, and shares its estimate of $\overline{\mathbf{f}}_i$ with neighbors in $\mathcal{N}_i$. By modifying Equation (98), we formulate the following:

$$\begin{aligned} \overline{\mathbf{f}}_i^{k+1} &= \underset{\overline{\mathbf{f}}_i}{\operatorname{argmin}} \frac{1}{2} \left(\sum_{j \in \mathcal{N}_i} \left\| \overline{\mathbf{f}}_i - \overline{\mathbf{f}}_j^k \right\|^2 \right) \\ \text{subject to} \quad & d_i = \mathbf{L}_i \overline{\mathbf{f}}_i. \end{aligned} \tag{101}$$

We solve Equation (101) using the Lagrange multiplier method such that

$$\overline{\mathbf{f}}_i^{k+1} = \underset{\overline{\mathbf{f}}_i}{\operatorname{argmin}} \left[\frac{1}{2} \left(\sum_{j \in \mathcal{N}_i} \left\| \overline{\mathbf{f}}_i - \overline{\mathbf{f}}_j^k \right\|^2 \right) + \lambda \left(d_i - \mathbf{L}_i \overline{\mathbf{f}}_i \right) \right],$$

where λ is the Lagrange multiplier. By solving this optimization problem, we obtain

$$\overline{\mathbf{f}}_i^{k+1} = \frac{\sum_{j\in\mathcal{N}_i}\overline{\mathbf{f}}_j^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_i - \mathbf{L}_i\sum_{j\in\mathcal{N}_i}\overline{\mathbf{f}}_j^k}{|\mathcal{N}_i|\|\mathbf{L}_i\|^2}\mathbf{L}_i^T, \tag{102}$$

where the first term on the right side of the equation is the average of the collected estimates and the second is the adjustment of the estimate for node v_i with respect to the collected estimates. This update equation represents the projection of the average of the collected estimates at the kth communication round onto the hyperplane $\mathcal{H}_i = \{\overline{\mathbf{f}} \in \mathbb{R}^n | \mathbf{L}_i\overline{\mathbf{f}} = d_i\}$. Considering that each node has the knowledge of its own d_i and $\mathbf{L}_i$, which form its own hyperplane, by sharing the estimate back and forth with neighbors, each node obtains the estimate projected onto the hyperplane of neighboring nodes.

6.2.1 Convergence Analysis

For any matrix $\mathbf{L}$ whose ith row is $\mathbf{L}_i$, let us define $\mathbf{L}_i^+ \equiv \frac{\mathbf{L}_i^T}{\|\mathbf{L}_i\|^2}$, referred to as the pseudo-inverse of $\mathbf{L}_i$.

Lemma 4. *$\mathbf{L}_i^+$ satisfies the following four conditions for the Moore-Penrose pseudo-inverse [67]:*

1. $\mathbf{L}_i\mathbf{L}_i^+\mathbf{L}_i = \mathbf{L}_i$
2. $\mathbf{L}_i^+\mathbf{L}_i\mathbf{L}_i^+ = \mathbf{L}_i^+$
3. $(\mathbf{L}_i\mathbf{L}_i^+)^T = \mathbf{L}_i\mathbf{L}_i^+$
4. $(\mathbf{L}_i^+\mathbf{L}_i)^T = \mathbf{L}_i^+\mathbf{L}_i$

To prove the convergence of $\overline{\mathbf{f}}_i^k$ updated by Equation (102), we make the following assumption.

Assumption 8. *Given any real number $0 < \beta < 1$ that applies for all $i = \{1, \cdots, K\}$ and a true solution $\overline{\mathbf{f}}_i^\star$ to Equation (100), there exists a ball $\mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$ around $\overline{\mathbf{f}}_i^\star$*

with radius $\delta > 0$ *such that the following holds for all* $\overline{\mathbf{f}}_i \in \mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$. *For a sequence* $\overline{\mathbf{f}}_i^k$ *generated by Equation* (102), *let* $\mathbf{e}_i^k = \overline{\mathbf{f}}_i^k - \overline{\mathbf{f}}_i^\star$ *and* $M_i = I - \mathbf{L}_i^+ \mathbf{L}_i$. *Then, for every* h *iterations with a small positive integer* h, *there exists at least one* $k \in \{nh, nh+1, \cdots, (n+1)h-1\}$, $n = \{0, 1, 2, \cdots\}$ *such that* $\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k$ *satisfies* $\left\langle M_i \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle \leq (1-\beta) \left\langle \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle$ *where* $\mathcal{N}_i$ *is a set of neighbors of node* v_i.

Theorem 6. *Suppose Assumption 8 holds for Equation* (100) *and its solution* $\overline{\mathbf{f}}_i^\star$. *Starting from any initial point* $\overline{\mathbf{f}}_i^0$ *that satisfies* $\|\overline{\mathbf{f}}_i^0 - \overline{\mathbf{f}}_i^\star\| < \nu$, $\nu > 0$ *within a ball* $\mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$, *the sequence* $\overline{\mathbf{f}}_i^k$ *generated by Equation* (102) *converges to* $\overline{\mathbf{f}}_i^\star$ *as* $k \to \infty$.

Proof. By subtracting $\overline{\mathbf{f}}_i^\star$ from both sides of the equation and substituting $\mathbf{L}_i^+$, Equation (102) becomes

$$\overline{\mathbf{f}}_i^{k+1} - \overline{\mathbf{f}}_i^\star = \frac{\sum_{j \in \mathcal{N}_i} \overline{\mathbf{f}}_j^k}{|\mathcal{N}_i|} - \overline{\mathbf{f}}_i^\star + \mathbf{L}_i^+ \frac{|\mathcal{N}_i| d_i - \mathbf{L}_i \sum_{j \in \mathcal{N}_i} \overline{\mathbf{f}}_j^k}{|\mathcal{N}_i|}.$$

By substituting $d_i = \mathbf{L}_i \overline{\mathbf{f}}_i^\star$, the equation becomes

$$\overline{\mathbf{f}}_i^{k+1} - \overline{\mathbf{f}}_i^\star = \frac{\sum_{j \in \mathcal{N}_i} \overline{\mathbf{f}}_j^k - |\mathcal{N}_i| \overline{\mathbf{f}}_i^\star}{|\mathcal{N}_i|} - \mathbf{L}_i^+ \frac{\mathbf{L}_i \sum_{j \in \mathcal{N}_i} \overline{\mathbf{f}}_j^k - |\mathcal{N}_i| \mathbf{L}_i \overline{\mathbf{f}}_i^\star}{|\mathcal{N}_i|}. \tag{103}$$

Let us define $\mathbf{e}_i^k = \overline{\mathbf{f}}_i^k - \overline{\mathbf{f}}_i^\star$ and simplify

$$\begin{aligned} \sum_{j \in \mathcal{N}_i} \overline{\mathbf{f}}_j^k - |\mathcal{N}_i| \overline{\mathbf{f}}_i^\star &= \sum_{j \in \mathcal{N}_i} \left(\overline{\mathbf{f}}_j^k - \overline{\mathbf{f}}_i^\star \right) \\ &= \sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k. \end{aligned} \tag{104}$$

By substituting $\mathbf{e}_i^k$ and Equation (104) into Equation (103), we obtain

$$\mathbf{e}_i^{k+1} = \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} - \mathbf{L}_i^+ \mathbf{L}_i \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}. \tag{105}$$

By taking the squared norm and using the property of the inner product, Equation (105) becomes

$$\|\mathbf{e}_i^{k+1}\|^2 = \left\langle \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle + \left\langle \left(\mathbf{L}_i^+ \mathbf{L}_i \right)^T \mathbf{L}_i^+ \mathbf{L}_i \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j \in \mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle$$

$$- 2 \left\langle \mathbf{L}_i^+ \mathbf{L}_i \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle. \tag{106}$$

By using Lemma 4 and simplifying the equation, Equation (106) becomes

$$\|\mathbf{e}_i^{k+1}\|^2 = \left\langle \left(I - \mathbf{L}_i^+ \mathbf{L}_i\right) \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle. \tag{107}$$

Let us define $M_i = I - \mathbf{L}_i^+ \mathbf{L}_i$. By the definition of $\mathbf{L}_i^+$, $\mathbf{L}_i^+ \mathbf{L}_i$ is normalized and it is a rank-one Hermitian matrix with the only non-zero eigenvalue being one. Therefore, M_i is positive semidefinite with the largest eigenvalue being one (c.f., [37], Theorem 4.3.1 (Weyl)). By Assumption 8, there are the following two cases:

Case 1.

$$\|\mathbf{e}_i^{k+1}\|^2 \leq \left\| \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\|^2,$$

or equivalently,

$$\|\mathbf{e}_i^{k+1}\| \leq \left\| \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\|.$$

Case 2.

$$\|\mathbf{e}_i^{k+1}\|^2 \leq (1-\beta) \left\| \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\|^2,$$

or equivalently,

$$\|\mathbf{e}_i^{k+1}\| \leq \sqrt{1-\beta} \left\| \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\|.$$

In the worst case, Case 2 happens only once and the rest corresponds to Case 1 for $k \in \{nh, nh+1, \cdots, (n+1)h-1\}$, $n = \{0, 1, 2, \cdots\}$. Without loss of generality, we can consider that Case 2 happens first and the rest falls into Case 1 for each h iterations. For $k = 0$, since $\|\overline{\mathbf{f}}_i^0 - \overline{\mathbf{f}}_i^\star\| < \nu$ for all i by assumption, we obtain

$$\begin{aligned} \|\mathbf{e}_i^1\| &\leq \sqrt{1-\beta} \left\| \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^0}{|\mathcal{N}_i|} \right\| \\ &\leq \sqrt{1-\beta} \frac{\sum_{j\in\mathcal{N}_i} \left\|\mathbf{e}_j^0\right\|}{|\mathcal{N}_i|} \\ &< \left(\sqrt{1-\beta}\right) \nu. \end{aligned} \tag{108}$$

Since Equation (108) holds for all i, for $k = 1$, we have

$$\begin{aligned}\|\mathbf{e}_i^2\| &\leq \left\| \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^1}{|\mathcal{N}_i|} \right\| \\ &\leq \frac{\sum_{j\in\mathcal{N}_i} \left\|\mathbf{e}_j^1\right\|}{|\mathcal{N}_i|} \\ &< \left(\sqrt{1-\beta}\right)\nu.\end{aligned}$$

By induction, we have for $k = h - 1$,

$$\|\mathbf{e}_i^h\| < \left(\sqrt{1-\beta}\right)\nu,$$

and for $k = nh - 1$,

$$\|\mathbf{e}_i^{nh}\| < \left(\sqrt{1-\beta}\right)^n \nu,$$

Since $\sqrt{1-\beta} < 1$ by Assumption 8, $\|\mathbf{e}_i^{nh}\|$ converges to 0 as $n \to \infty$. Therefore, $\|\mathbf{e}_i^k\|$ converges to 0 as $k \to \infty$, and we conclude that $\overline{\mathbf{f}}_i^k$ converges to $\overline{\mathbf{f}}_i^\star$ in the L2 sense for all i. This proves the theorem. □

6.3 A Distributed Kaczmarz-Type Method for MT

This section presents a distributed Kaczmarz-type method [10] for solving a nonlinear system of equations constructed for MT, such as Equations (20), (51), and (71). For simplicity of presentation, let us omit x and y in the system of equations and consider

$$\mathbf{d} = \mathbf{L}(\overline{\mathbf{f}})\overline{\mathbf{f}}, \tag{109}$$

where

$$\mathbf{L}(\overline{\mathbf{f}}) = \begin{bmatrix} \mathbf{L}_1(\overline{\mathbf{f}}) \\ \vdots \\ \mathbf{L}_K(\overline{\mathbf{f}}) \end{bmatrix}, \ \overline{\mathbf{f}} = \begin{bmatrix} f_1 \\ \vdots \\ f_M \end{bmatrix}, \ \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix}.$$

To solve Equation (109) in a distributed fashion, we consider that node v_i solves

$$d_i = \mathbf{L}_i(\overline{\mathbf{f}}_i)\overline{\mathbf{f}}_i, \tag{110}$$

where $\mathbf{L}_i$ is the ith row of matrix $\mathbf{L}$ and d_i the ith element of vector $\mathbf{d}$, and shares its estimate of $\overline{\mathbf{f}}_i$ with neighbors in $\mathcal{N}_i$. Then, we modify Equation (32) and formulate the following:

$$\begin{aligned} \overline{\mathbf{f}}_i^{k+1} = \underset{\overline{\mathbf{f}}_i}{\operatorname{argmin}} \frac{1}{2} \left(\sum_{j \in \mathcal{N}_i} \left\| \overline{\mathbf{f}}_i - \overline{\mathbf{f}}_j^k \right\|^2 \right) \\ \text{subject to} \qquad d_i = \mathbf{L}_i(\overline{\mathbf{f}}_i^k) \overline{\mathbf{f}}_i, \end{aligned} \tag{111}$$

where $k = \{0, 1, 2, \cdots\}$ is the index for iterations or communication rounds. By solving Equation (111) using the Lagrange multiplier method, we obtain

$$\overline{\mathbf{f}}_i^{k+1} = \frac{\sum_{j \in \mathcal{N}_i} \overline{\mathbf{f}}_j^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i| d_i - \mathbf{L}_i(\overline{\mathbf{f}}_i^k) \sum_{j \in \mathcal{N}_i} \overline{\mathbf{f}}_j^k}{|\mathcal{N}_i| \|\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\|^2} (\mathbf{L}_i(\overline{\mathbf{f}}_i^k))^T. \tag{112}$$

As is the case with the distributed Kaczmarz method presented in the previous section, this update equation represents the projection of the average of the collected estimates at the kth communication round onto the hyperplane $\mathcal{H}_i = \{\overline{\mathbf{f}} \in \mathbb{R}^n | \mathbf{L}_i(\overline{\mathbf{f}}_i^k) \overline{\mathbf{f}} = d_i\}$. Considering that each node has the knowledge of its own d_i and $\mathbf{L}_i$, which form its own hyperplane, by sharing the estimate back and forth with neighbors, each node obtains the estimate projected onto the hyperplane of neighboring nodes.

6.3.1 Convergence Analysis

For any matrix $\mathbf{L}(\overline{\mathbf{f}})$ whose ith row is $\mathbf{L}_i(\overline{\mathbf{f}})$, let us define the pseudo-inverse of $\mathbf{L}_i(\overline{\mathbf{f}})$ as $\mathbf{L}_i^+(\overline{\mathbf{f}}) \equiv \frac{(\mathbf{L}_i(\overline{\mathbf{f}}))^T}{\|\mathbf{L}_i(\overline{\mathbf{f}})\|^2}$.

Lemma 5. $\mathbf{L}_i^+(\overline{\mathbf{f}})$ *satisfies the conditions for the Moore-Penrose pseudo-inverse [67].*

To prove the convergence of $\overline{\mathbf{f}}_i^k$ updated by Equation (112), we assume that there exists a ball $\mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$ around a true solution $\overline{\mathbf{f}}_i^\star$ with radius $\delta > 0$ where the following two assumptions hold for all $\overline{\mathbf{f}}_i \in \mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$, $i = \{1, \cdots, K\}$:

Assumption 9. $\mathbf{L}_i(\overline{\mathbf{f}}_i)$ *in Equation* (110) *is Lipschitz continuous for all* $i = \{1, \cdots, K\}$ *with the largest Lipschitz constant* $\gamma_L = \max_i \gamma_L^i$*, where* γ_L^i *is the Lipschitz constant for* $\mathbf{L}_i(\overline{\mathbf{f}})$*.*

Assumption 10. *Given any real numbers $0 < \epsilon, \beta < 1$ that applies for all $i = \{1, \cdots, K\}$ and a true solution $\overline{\mathbf{f}}_i^\star$ to Equation (110), there exists a ball $\mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$ around $\overline{\mathbf{f}}_i^\star$ with radius $\delta > 0$ such that the following hold for all $\bar{\mathbf{f}}_i \in \mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$:*

1) $\dfrac{\gamma_L \|\overline{\mathbf{f}}_i^\star\|}{\|\mathbf{L}_i(\bar{\mathbf{f}}_i)\|} < \sqrt{\epsilon}$ *for all* $i = \{1, \cdots, K\}$.

2) *For a sequence $\overline{\mathbf{f}}_i^k$ generated by Equation (112), let $\mathbf{e}_i^k = \overline{\mathbf{f}}_i^k - \overline{\mathbf{f}}_i^\star$ and $M_i^k = I - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)$. Then, for every h iterations with a small positive integer h, there exists at least one $k \in \{nh, nh+1, \cdots, (n+1)h-1\}$, $n = \{0, 1, 2, \cdots\}$ such that $\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k$ satisfies $\left\langle M_i^k \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle \leq (1-\beta) \left\langle \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle$ where $\mathcal{N}_i$ is a set of neighbors of node v_i and $\beta > 1 + \epsilon - \dfrac{1}{(1+\epsilon)^{h-1}}$.*

Theorem 7. *Suppose Assumptions 9 and 10 hold for Equation (110) and its solution $\overline{\mathbf{f}}_i^\star$. Starting from any initial point $\overline{\mathbf{f}}_i^0$ that satisfies $\|\overline{\mathbf{f}}_i^0 - \overline{\mathbf{f}}_i^\star\| < \nu$, $\nu > 0$ within a ball $\mathcal{B}(\overline{\mathbf{f}}_i^\star, \delta)$, the sequence $\overline{\mathbf{f}}_i^k$ generated by Equation (112) converges to $\overline{\mathbf{f}}_i^\star$ as $k \to \infty$.*

Proof. By subtracting $\overline{\mathbf{f}}_i^\star$ from both sides of Equation (112) and substituting $\mathbf{L}_i^+(\overline{\mathbf{f}})$, Equation (112) becomes

$$\overline{\mathbf{f}}_i^{k+1} - \overline{\mathbf{f}}_i^\star = \frac{\sum_{j\in\mathcal{N}_i} \overline{\mathbf{f}}_j^k - |\mathcal{N}_i|\overline{\mathbf{f}}_i^\star}{|\mathcal{N}_i|} - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k) \frac{\mathbf{L}_i(\overline{\mathbf{f}}_i^k) \sum_{j\in\mathcal{N}_i} \overline{\mathbf{f}}_j^k - |\mathcal{N}_i| d_i}{|\mathcal{N}_i|}. \tag{113}$$

Let us define $\mathbf{e}_i^k = \overline{\mathbf{f}}_i^k - \overline{\mathbf{f}}_i^\star$. Since $d_i = \mathbf{L}_i(\overline{\mathbf{f}}_i^\star)\overline{\mathbf{f}}_i^\star$, we can express $\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\overline{\mathbf{f}}_j^k - d_i$ as

$$\begin{aligned}
\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\overline{\mathbf{f}}_j^k - d_i &= \mathbf{L}_i(\overline{\mathbf{f}}_i^k)\overline{\mathbf{f}}_j^k - \mathbf{L}_i(\overline{\mathbf{f}}_i^\star)\overline{\mathbf{f}}_i^\star \\
&= \mathbf{L}_i(\overline{\mathbf{f}}_i^k)\overline{\mathbf{f}}_j^k - \mathbf{L}_i(\overline{\mathbf{f}}_i^k)\overline{\mathbf{f}}_i^\star + \mathbf{L}_i(\overline{\mathbf{f}}_i^k)\overline{\mathbf{f}}_i^\star - \mathbf{L}_i(\overline{\mathbf{f}}_i^\star)\overline{\mathbf{f}}_i^\star \\
&= \mathbf{L}_i(\overline{\mathbf{f}}_i^k)\mathbf{e}_j^k + \mathbf{p}_i^k \overline{\mathbf{f}}_i^\star,
\end{aligned} \tag{114}$$

where $\mathbf{p}_i^k = \mathbf{L}_i(\overline{\mathbf{f}}_i^k) - \mathbf{L}_i(\overline{\mathbf{f}}_i^\star)$. Then, we simplify

$$\begin{aligned}
\sum_{j\in\mathcal{N}_i} \overline{\mathbf{f}}_j^k - |\mathcal{N}_i|\overline{\mathbf{f}}_i^\star &= \sum_{j\in\mathcal{N}_i} \left(\overline{\mathbf{f}}_j^k - \overline{\mathbf{f}}_i^\star\right) \\
&= \sum_{j\in\mathcal{N}_i} \mathbf{e}_j^k,
\end{aligned} \tag{115}$$

and

$$\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\sum_{j\in\mathcal{N}_i}\overline{\mathbf{f}}_j^k - |\mathcal{N}_i|d_i = \sum_{j\in\mathcal{N}_i}\Big(\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\overline{\mathbf{f}}_j^k - d_i\Big)$$
$$= \sum_{j\in\mathcal{N}_i}(\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\mathbf{e}_j^k + \mathbf{p}_i^k\overline{\mathbf{f}}_i^\star). \tag{116}$$

By substituting $\mathbf{e}_i^k$ and Equations (115) and (116) into Equation (113), we obtain

$$\mathbf{e}_i^{k+1} = \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star. \tag{117}$$

By taking the squared norm and expanding the equation, Equation (117) becomes

$$\begin{aligned}\|\mathbf{e}_i^{k+1}\|^2 = &\left\langle \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle - 2\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle \\ &- 2\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle + 2\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle \\ &+ \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle \\ &+ \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star\right\rangle \end{aligned} \tag{118}$$

By the property of the inner product and Lemma 5, the fourth and fifth terms on the right side of Equation (118) become

$$\begin{aligned}\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle &= \left\langle \Big(\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\Big)^T\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle \\ &= \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle \end{aligned} \tag{119}$$

and

$$\begin{aligned}&\left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle \\ &\quad= \left\langle \Big(\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\Big)^T\mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle \\ &\quad= \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}^k)\mathbf{L}_i(\overline{\mathbf{f}}^k)\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\rangle, \end{aligned} \tag{120}$$

respectively. By substituting Equations (119) and (120) into Equation (118), we have

$$\begin{aligned}\|\mathbf{e}_i^{k+1}\|^2 &= \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\right) \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle + \left\langle \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star, \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star \right\rangle \\ &= \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\right) \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle + \left\| \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{p}_i^k\overline{\mathbf{f}}_i^\star \right\|^2.\end{aligned}$$

Since $\mathbf{L}_i(\overline{\mathbf{f}})$ is Lipschitz continuous, we have

$$\begin{aligned}\|\mathbf{p}_i^k\| &= \|\mathbf{L}_i(\overline{\mathbf{f}}_i^k) - \mathbf{L}_i(\overline{\mathbf{f}}_i^\star)\| \\ &\leq \gamma_L\|\overline{\mathbf{f}}_i^k - \overline{\mathbf{f}}_i^\star\| \\ &= \gamma_L\|\mathbf{e}_i^k\|,\end{aligned}$$

in which γ_L is the Lipschitz constant defined in Assumption 9. Then, we obtain

$$\|\mathbf{e}_i^{k+1}\|^2 \leq \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\right) \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle + \gamma_L^2\|\mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\|^2\|\overline{\mathbf{f}}_i^\star\|^2\|\mathbf{e}_i^k\|^2$$

By using $\|(\mathbf{L}_i(\overline{\mathbf{f}}^k))^+\|^2 = \left\| \frac{\mathbf{L}_i(\overline{\mathbf{f}}^k)^T}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2} \right\|^2 = \frac{1}{\|\mathbf{L}_i(\overline{\mathbf{f}}^k)\|^2}$ and Assumption 10, we obtain

$$\begin{aligned}\|\mathbf{e}_i^{k+1}\|^2 &\leq \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\right) \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle + \frac{\gamma_L^2\|\overline{\mathbf{f}}_i^\star\|^2}{\|\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\|^2}\|\mathbf{e}_i^k\|^2 \\ &< \left\langle \left(I - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\right) \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}, \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} \right\rangle + \epsilon\|\mathbf{e}_i^k\|^2.\end{aligned}$$

Let us define $M_i^k = I - \mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)$. By the definition of $\mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)$, $\mathbf{L}_i^+(\overline{\mathbf{f}}_i^k)\mathbf{L}_i(\overline{\mathbf{f}}_i^k)$ is normalized and it is a rank-one Hermitian matrix with the only non-zero eigenvalue being one. Therefore, M_i^k is positive semidefinite with the largest eigenvalue being one (c.f., [37], Theorem 4.3.1 (Weyl)). By Assumption 10, there are the following two cases:

Case 1.

$$\|\mathbf{e}_i^{k+1}\|^2 < \left\| \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} \right\|^2 + \epsilon\|\mathbf{e}_i^k\|^2,$$

or equivalently,

$$\|\mathbf{e}_i^{k+1}\| < \sqrt{\left\| \frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|} \right\|^2 + \epsilon\|\mathbf{e}_i^k\|^2}.$$

Case 2.

$$\|\mathbf{e}_i^{k+1}\|^2 < (1-\beta)\left\|\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\|^2 + \epsilon\|\mathbf{e}_i^k\|^2,$$

or equivalently,

$$\|\mathbf{e}_i^{k+1}\| < \sqrt{(1-\beta)\left\|\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^k}{|\mathcal{N}_i|}\right\|^2 + \epsilon\|\mathbf{e}_i^k\|^2}.$$

In the worst case, Case 2 happens only once and the rest corresponds to Case 1 for $k \in \{nh, nh+1, \cdots, (n+1)h-1\}$, $n = \{0, 1, 2, \cdots\}$. Without loss of generality, we can consider that Case 2 happens first and the rest falls into Case 1 for each h iterations. For $k = 0$, since $\|\overline{\mathbf{f}}_i^0 - \overline{\mathbf{f}}_i^\star\| < \nu$ for all i by assumption, we obtain

$$\begin{aligned}
\|\mathbf{e}_i^1\| &< \sqrt{(1-\beta)\left\|\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^0}{|\mathcal{N}_i|}\right\|^2 + \epsilon\|\mathbf{e}_i^0\|^2} \\
&\leq \sqrt{(1-\beta)\left(\frac{\sum_{j\in\mathcal{N}_i}\left\|\mathbf{e}_j^0\right\|}{|\mathcal{N}_i|}\right)^2 + \epsilon\|\mathbf{e}_i^0\|^2} \\
&< \left(\sqrt{1-\beta+\epsilon}\right)\nu. \qquad (121)
\end{aligned}$$

Since Equation (121) holds for all i, for $k = 1$, we have

$$\begin{aligned}
\|\mathbf{e}_i^2\| &< \sqrt{\left\|\frac{\sum_{j\in\mathcal{N}_i}\mathbf{e}_j^1}{|\mathcal{N}_i|}\right\|^2 + \epsilon\|\mathbf{e}_i^1\|^2} \\
&\leq \sqrt{\left(\frac{\sum_{j\in\mathcal{N}_i}\left\|\mathbf{e}_j^1\right\|}{|\mathcal{N}_i|}\right)^2 + \epsilon\|\mathbf{e}_i^1\|^2} \\
&< \sqrt{\left(\frac{\sum_{j\in\mathcal{N}_i}\left(\sqrt{1-\beta+\epsilon}\right)\nu}{|\mathcal{N}_i|}\right)^2 + \epsilon\left(\left(\sqrt{1-\beta+\epsilon}\right)\nu\right)^2} \\
&= \sqrt{(1-\beta+\epsilon)\nu^2 + \epsilon(1-\beta+\epsilon)\nu^2} \\
&= \left(\sqrt{1-\beta+\epsilon}\right)\left(\sqrt{1+\epsilon}\right)\nu.
\end{aligned}$$

By induction, we have for $k = h-1$,

$$\|\mathbf{e}_i^h\| < \left(\sqrt{1-\beta+\epsilon}\right)\left(\sqrt{1+\epsilon}\right)^{h-1}\nu,$$

and for $k = nh - 1$,

$$\|\mathbf{e}_i^{nh}\| < \left(\left(\sqrt{1-\beta+\epsilon}\right)\left(\sqrt{1+\epsilon}\right)^{h-1}\right)^n \nu,$$

Since $\left(\sqrt{1-\beta+\epsilon}\right)\left(\sqrt{1+\epsilon}\right)^{h-1} < 1$ by Assumption 10, $\|\mathbf{e}_i^{nh}\|$ converges to 0 as $n \to \infty$. Therefore, $\|\mathbf{e}_i^{k}\|$ converges to 0 as $k \to \infty$ and we conclude that $\overline{\mathbf{f}}_i^k$ converges to $\overline{\mathbf{f}}_i^\star$ in the L2 sense for all i. This proves the theorem. □

6.4 Distributed Flow Field Estimation by MT

Given a group of K vehicles, suppose they form a static network after they finish their navigation over one observation interval. Let us consider that each vehicle has knowledge of its own motion-integration error $\mathbf{d}_i$ and trajectory information $\mathbf{L}_i$, $i = \{1, \cdots, K\}$. In other words, each vehicle forms its own motion-integration error equations

$$\begin{aligned} d_{i,x} &= \mathbf{L}_i(\overline{\mathbf{f}})\overline{\mathbf{f}}_x \\ d_{i,y} &= \mathbf{L}_i(\overline{\mathbf{f}})\overline{\mathbf{f}}_y. \end{aligned} \tag{122}$$

To estimate a flow field through MT in a distributed fashion [10], the ith vehicle shares its estimate of a flow field with its neighbors in $\mathcal{N}_i$ and iteratively updates its own estimate of the solution to its own motion-integration error equation based on Equation (112). Algorithm 5 shows the implementation of the flow field estimation for the non-parametric version of distributed MT.

Considering the dimension of the solution to Equation (122), sharing this solution may have high communication cost. Therefore, to reduce the dimension of the solution variable, a data-driven flow model with basis functions and corresponding parameters can be incorporated. In other words, we estimate a flow field through parametric MT in a distributed fashion. Suppose at each iteration k, vehicle i obtains Θ_i^k and Θ_j^k, $j \in \mathcal{N}_i$ for parametric MT. Then, to implement the parametric version of distributed MT, we modify the sub-optimization problem in Equations (81) and (83) for parametric

Algorithm 5: Distributed MT flow field estimation

Data: Motion-integration errors $\mathbf{d}_i, i = \{1, \cdots, K\}$

Set $k = 0$.

Make an initial guess of the solutions, $\overline{\mathbf{f}}_{x,i}^0$ and $\overline{\mathbf{f}}_{y,i}^0$.

repeat

  Vehicle v_i shares its estimates $\overline{\mathbf{f}}_{x,i}^k$ and $\overline{\mathbf{f}}_{y,i}^k$ with its neighbors $v_j \in \mathcal{N}_i$.

  Each vehicle updates $\overline{\mathbf{f}}_{x,i}^{k+1}$ and $\overline{\mathbf{f}}_{y,i}^{k+1}$ by

$$\overline{\mathbf{f}}_{x,i}^{k+1} = \frac{\sum_{j\in\mathcal{N}_i} \overline{\mathbf{f}}_{x,j}^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i| d_{x,i} - \mathbf{L}_i(\overline{\mathbf{f}}_i^k) \sum_{j\in\mathcal{N}_i} \overline{\mathbf{f}}_{x,j}^k}{|\mathcal{N}_i| \|\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\|^2} (\mathbf{L}_i(\overline{\mathbf{f}}_i^k))^T$$
$$\overline{\mathbf{f}}_{y,i}^{k+1} = \frac{\sum_{j\in\mathcal{N}_i} \overline{\mathbf{f}}_{y,j}^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i| d_{y,i} - \mathbf{L}_i(\overline{\mathbf{f}}_i^k) \sum_{j\in\mathcal{N}_i} \overline{\mathbf{f}}_{y,j}^k}{|\mathcal{N}_i| \|\mathbf{L}_i(\overline{\mathbf{f}}_i^k)\|^2} (\mathbf{L}_i(\overline{\mathbf{f}}_i^k))^T.$$

  Let $k = k + 1$.

until *a stopping condition is met*

MT and formulate the following optimization problems:

$$\begin{aligned} \eta_{x,i}^{k+1} &= \underset{\eta_{x,i}}{\operatorname{argmin}} \frac{1}{2} \left(\sum_{j\in\mathcal{N}_i} \left\| \eta_{x,i} - \eta_{x,j}^k \right\|^2 \right) \\ \eta_{y,i}^{k+1} &= \underset{\eta_{y,i}}{\operatorname{argmin}} \frac{1}{2} \left(\sum_{j\in\mathcal{N}_i} \left\| \eta_{y,i} - \eta_{y,j}^k \right\|^2 \right) \end{aligned} \tag{123}$$

subject to

$$\begin{aligned} d_{x,i} &= \mathbf{G}_i^{\rho_{x,i}^k}(\eta_i^k)\eta_{x,i} \\ d_{y,i} &= \mathbf{G}_i^{\rho_{y,i}^k}(\eta_i^k)\eta_{y,i}, \end{aligned} \tag{124}$$

and

$$\begin{aligned} \rho_{x,i}^{k+1} &= \underset{\rho_{x,i}}{\operatorname{argmin}} \frac{1}{2} \left(\sum_{j\in\mathcal{N}_i} \left\| \rho_{x,i} - \rho_{x,j}^k \right\|^2 \right) \\ \rho_{y,i}^{k+1} &= \underset{\rho_{y,i}}{\operatorname{argmin}} \frac{1}{2} \left(\sum_{j\in\mathcal{N}_i} \left\| \rho_{y,i} - \rho_{y,j}^k \right\|^2 \right) \end{aligned} \tag{125}$$

subject to

$$\begin{aligned} d_{x,i} &= \mathbf{G}_i^{\eta_{x,i}^k}(\rho_i^k)\rho_{x,i} \\ d_{y,i} &= \mathbf{G}_i^{\eta_{y,i}^k}(\rho_i^k)\rho_{y,i}. \end{aligned} \tag{126}$$

By solving the optimization problems in Equations (123) and (125), at each iteration, we update the solutions by the following equations:

$$\eta_{x,i}^{k+1} = \frac{\sum_{j\in\mathcal{N}_i}\eta_{x,j}^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{x,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\rho_{x,i}^k}(\eta_i^k)\eta_{x,j}^k}{|\mathcal{N}_i|\|\mathbf{G}_i^{\rho_{x,i}^k}(\eta_i^k)\|^2}\left(\mathbf{G}_i^{\rho_{x,i}^k}(\eta_i^k)\right)^T \tag{127}$$

$$\eta_{y,i}^{k+1} = \frac{\sum_{j\in\mathcal{N}_i}\eta_{y,j}^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{y,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\rho_{y,i}^k}(\eta_i^k)\eta_{y,j}^k}{|\mathcal{N}_i|\|\mathbf{G}_i^{\rho_{y,i}^k}(\eta_i^k)\|^2}\left(\mathbf{G}_i^{\rho_{y,i}^k}(\eta_i^k)\right)^T, \tag{128}$$

and

$$\rho_{x,i}^{k+1} = \frac{\sum_{j\in\mathcal{N}_i}\rho_{x,j}^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{x,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\eta_{x,i}^k}(\rho_i^k)\rho_{x,j}^k}{|\mathcal{N}_i|\|\mathbf{G}_i^{\eta_{x,i}^k}(\rho_i^k)\|^2}\left(\mathbf{G}_i^{\eta_{x,i}^k}(\rho_i^k)\right)^T \tag{129}$$

$$\rho_{y,i}^{k+1} = \frac{\sum_{j\in\mathcal{N}_i}\rho_{y,j}^k}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{y,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\eta_{y,i}^k}(\rho_i^k)\rho_{y,j}^k}{|\mathcal{N}_i|\|\mathbf{G}_i^{\eta_{y,i}^k}(\rho_i^k)\|^2}\left(\mathbf{G}_i^{\eta_{y,i}^k}(\rho_i^k)\right)^T, \tag{130}$$

where the first term on the right side of each equation is the average of the estimates of the vehicles in the neighbor set and the second term is the adjustment of the average estimate with respect to all the available estimates based on the local motion-integration error equation.

Based on the distributed optimization problems designed above, we obtain a distributed parameter estimation algorithm (Algorithm 6). At the beginning, each vehicle computes initial guesses of estimates for flow model parameters independently. Then, at each iteration, vehicle i shares its estimate Θ_i^k with its neighbors $j \in \mathcal{N}_i$. Then, each vehicle updates their estimates using Equations (127)–(130). The algorithm repeats this process until a stopping condition is met.

6.5 Simulation Results for Time-Invariant Flow

This section validates the proposed distributed optimization method for solving a nonlinear system of equations formulated for MT. To demonstrate the method, we simulate multiple vehicles that navigate under flow in a domain of interest and implement the distributed method for MT to construct a map of an underlying flow field from trajectory information collected from the vehicles. The simulation setup

Algorithm 6: Distributed MT flow model parameter estimation

Data: Motion-integration errors $\mathbf{d}_i, i = \{1, \cdots, K\}$

Set $k = 0$.

Make an initial guess of the solutions, $\eta^0_{x,i}$, $\eta^0_{y,i}$, $\rho^0_{x,i}$, and $\rho^0_{y,i}$.

repeat

Each vehicle i shares its estimates $\eta^k_{x,i}$, $\eta^k_{y,i}$, $\rho^k_{x,i}$, and $\rho^k_{y,i}$ with its neighbors $j \in \mathcal{N}_i$.

Update the solutions by

$$\eta^{k+1}_{x,i} = \frac{\sum_{j\in\mathcal{N}_i}\eta^k_{x,j}}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{x,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\rho^k_{x,i}}(\eta^k_i)\eta^k_{x,j}}{|\mathcal{N}_i|\|\mathbf{G}_i^{\rho^k_{x,i}}(\eta^k_i)\|^2}\left(\mathbf{G}_i^{\rho^k_{x,i}}(\eta^k_i)\right)^T$$

$$\eta^{k+1}_{y,i} = \frac{\sum_{j\in\mathcal{N}_i}\eta^k_{y,j}}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{y,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\rho^k_{y,i}}(\eta^k_i)\eta^k_{y,j}}{|\mathcal{N}_i|\|\mathbf{G}_i^{\rho^k_{y,i}}(\eta^k_i)\|^2}\left(\mathbf{G}_i^{\rho^k_{y,i}}(\eta^k_i)\right)^T$$

$$\rho^{k+1}_{x,i} = \frac{\sum_{j\in\mathcal{N}_i}\rho^k_{x,j}}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{x,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\eta^{k+1}_{x,i}}(\rho^k_i)\rho^k_{x,j}}{|\mathcal{N}_i|\|\mathbf{G}_i^{\eta^{k+1}_{x,i}}(\rho^k_i)\|^2}\left(\mathbf{G}_i^{\eta^{k+1}_{x,i}}(\rho^k_i)\right)^T$$

$$\rho^{k+1}_{y,i} = \frac{\sum_{j\in\mathcal{N}_i}\rho^k_{y,j}}{|\mathcal{N}_i|} + \frac{|\mathcal{N}_i|d_{y,i} - \sum_{j\in\mathcal{N}_i}\mathbf{G}_i^{\eta^{k+1}_{y,i}}(\rho^k_i)\rho^k_{y,j}}{|\mathcal{N}_i|\|\mathbf{G}_i^{\eta^{k+1}_{y,i}}(\rho^k_i)\|^2}\left(\mathbf{G}_i^{\eta^{k+1}_{y,i}}(\rho^k_i)\right)^T.$$

Let $k = k + 1$.

Construct $\overline{\mathbf{f}}_{x,i}$ and $\overline{\mathbf{f}}_{y,i}$ using updated estimates for the parameters.

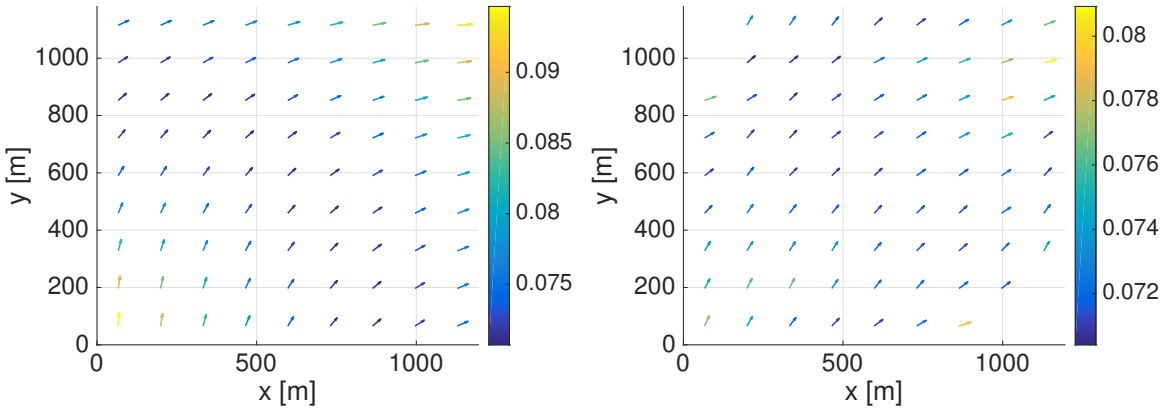
$\mathbf{r}^k_{\Theta,x,i} = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k_x - \mathbf{d}_x$.

$\mathbf{r}^k_{\Theta,y,i} = \mathbf{L}(\overline{\mathbf{f}}^k)\overline{\mathbf{f}}^k_y - \mathbf{d}_y$.

until *a stopping condition is met (e.g.,* $\|\mathbf{r}^k_{\overline{\mathbf{f}},x}\|, \|\mathbf{r}^k_{\overline{\mathbf{f}},y}\| \leq \epsilon_{\overline{\mathbf{f}}}$*)*

identical to that presented in Section 3.5.1 is constructed here, so the performance of MT through distributed computation for a time-invariant flow field can be compared to that of MT through non-distributed computation. A simulated "true" flow field (see Figure 22a) is constructed in a $1200\,m \times 1200\,m$ domain. As the distance from the current location to the origin increases, the x component of the flow velocity increases from 0 to $0.1\,m/s$ and the y component of the flow velocity decreases from $0.1\,m/s$ to 0.

Under the simulated flow field, we navigate $K = 18$ vehicles. Of the 18 vehicles, nine travel from the left of the domain to the right and the other nine from the



(a) A simulated "true" flow field.

(b) A constructed flow field.

Figure 22: A simulated "true" flow field where simulated vehicles travel and a constructed flow field by running the proposed method for distributed MT.

bottom to the top (see Figure 23). Vehicles travel at the horizontal through-water speed of $s_h = 0.35\,m/s$ until their estimated travel distance reaches $1000\,m$. Given a target point, the ith vehicle first computes heading θ_i towards the target point and moves straight forward towards it. To compute the predicted and actual trajectories of a vehicle, Equations (1) and (2) are discretized with time step $\Delta t = 1$ and the estimated flow in Equation (1) is assumed zero (i.e., $\tilde{\mathbf{f}} = \mathbf{0}$). We assume that the positions of the vehicles are known only at the starting and final times during their travel.

Once all the vehicles finish their travel, each vehicle computes its motion-integration error $\mathbf{d}_i$, $i = \{1, \cdots, K\}$. To implement distributed MT, a set of neighbors of the ith vehicle is determined by the distance between its final position $\mathbf{r}_i(t^f)$ and the final position of another vehicle $\mathbf{r}_j(t^f)$ (e.g., in the simulation $\mathcal{N}_i = \{v_j | \|\mathbf{r}_i(t^f) - \mathbf{r}_j(t^f)\| <= 300\,m\}$). Then, each vehicle updates its flow estimate by running Algorithm 5 based on the distributed method for MT.

Figure 22b displays the constructed flow field and Table 7 shows convergence conditions for the algorithms and mapping errors. To construct a map of a flow field

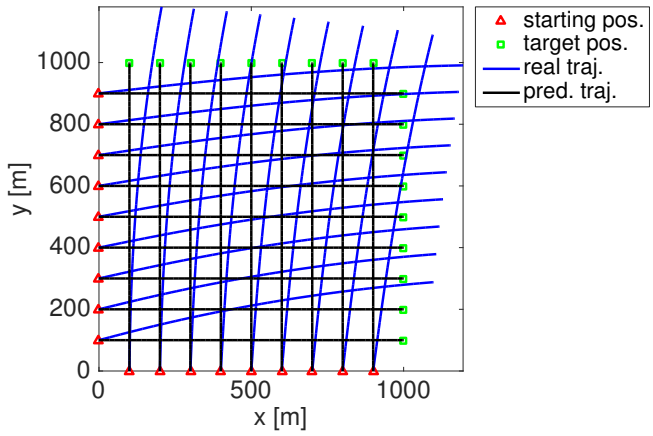


Figure 23: Predicted trajectories between starting positions (red triangles) and target positions (cyan rectangles), are displayed in black lines along with real trajectories in blue lines.

Table 7: Convergence conditions and quantitative results for non-parametric distributed MT simulations with time-invariant flow.

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$
$\epsilon_\gamma = 10^{-2}$	$\epsilon_{\overline{\mathbf{f}}} = 30$	$x : 0.0163\, m/s$ $y : 0.0148\, m/s$

through MT, the domain is discretized into $P = 9 \times 9$ grid cells. To compare the performance of distributed MT with non-distributed MT, we use the identical conditions for the flow field estimation algorithm as presented in Section 3.5.1. The rms errors in the x and y components for distributed MT with non-parametric time-invariant flow are 0.0163 m/s and 0.0148 m/s, respectively, and these numbers are comparable to the rms errors for non-distributed MT with non-parametric time-invariant flow presented in Table 1 in Section 3.5.1. These results demonstrate that the distributed method performs very closely to the non-distributed method and suggest that our distributed method successfully solves a nonlinear system of equations formulated for MT.

6.6 Simulation Results for Time-Varying Flow

This section validates the proposed methods through simulations under the same setup as in Section 5.1.3 (i.e., $K = 10$ vehicles traveling for $T_{obs} = 2$ hours, $P = 5 \times 5$ grid cells, $T = 5$ sub-time intervals, $N = 7$, $L = 3$, and $M = 3$). To simulate "true" flow, we use post-processed historic HF-radar data which provide approximately 3×3 km^2 spatial resolution and 30-minute temporal resolution. To parameterize the motion-integration errors in Equation (71), we use a data-driven computational flow model in Equation (6). To initialize basis functions for the flow model, we select 30-day historical HF-radar data from the time when vehicles start traveling.

After the basis functions for the flow model is initialized, we navigate $K = 10$ vehicles with horizontal speed $s_h = 0.35$ m/s for 2 hours in a domain. Of the $K = 10$ vehicles, 5 navigate from left to right and the other 5 from bottom to top. See Figure 16 for the visualization of simulated vehicle trajectories. This simulation setup is identical to the simulated presented in Section 5.1.3, so the performance of MT through distributed computation for a time-varying flow field can be compared to that of MT through non-distributed computation. Once vehicles complete their navigation, vehicles exchange their estimated parameters for the flow model in Equation (6) with their neighbors within 1 kilometer. Then, each vehicle updates its own parameter estimates using Algorithm 6 starting from $\eta^0_{x,i} = \eta^0_{y,i} = \rho^0_{x,i} = \rho^0_{y,i} = 0.1$, $i = \{1, \cdots, K\}$.

Figure 24 compares the x and y components of real flow at the center $(x, y) = (1260\,m, 1260\,m)$ of the domain and those of estimated flow constructed using parameters that are computed through distributed MT. From the figure, wee see that the estimated flow computed through distributed MT is very similar to that computed through non-distributed MT shown in Figure 17. It is intuitive that starting from the same initial condition, estimates of each vehicle updated using Equations (127)–(130) will reach the average of all the estimates and this average will converge to estimates

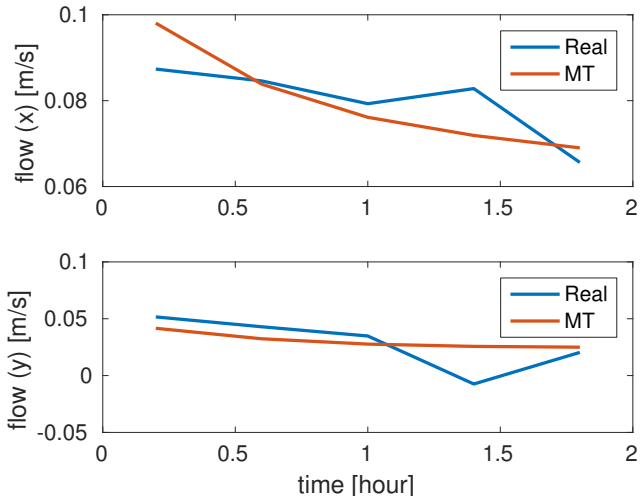


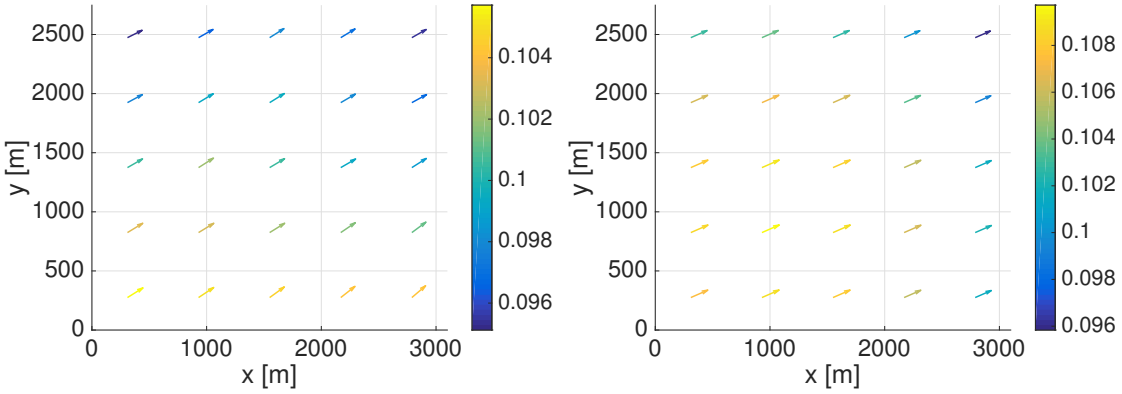
Figure 24: Comparison between real flow and estimated flow constructed using parameters computed through distributed MT at the center of the domain where vehicles navigate.

Table 8: Convergence conditions and quantitative results for parametric distributed MT simulations with time-varying flow at the center of the domain

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$
$\epsilon_\gamma = 10^{-2}$	$\epsilon_{\overline{\mathbf{f}}} = 30$	$x : 0.0072\, m/s$ $y : 0.0166\, m/s$

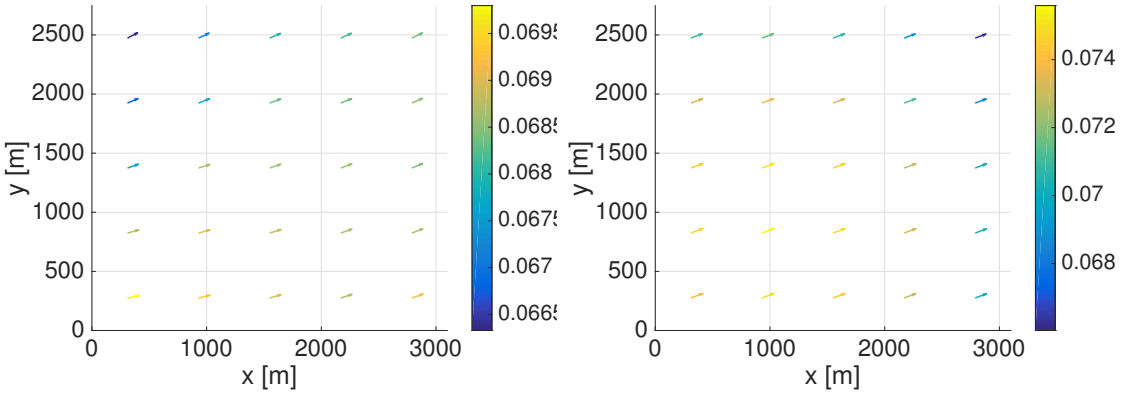
computed through non-distributed MT. Table 8 shows convergence conditions for the algorithms and mapping errors. The rms errors for the x and y components of the estimated flow obtained through distributed MT are the same as those for the x and y components of the estimated flow obtained through non-distributed MT.

Figure 25 compares "true" and estimated flow fields at 0.2 hour and at 1.8 hour, respectively. The range of the colorbar represents flow velocities. By comparing the range of the colorbar at two different times, we can see that the flow field is time-varying. Table 9 shows convergence conditions for the algorithms and mapping errors at two different times, $t = 0.2$ hour and $t = 1.8$ hour. The rms errors between "true" and estimated flow fields at 0.2 hour are 0.0115 m/s in the x direction and 0.0125



(a) True field at $t = 0.2$ hour.

(b) Estimated field at $t = 0.2$ hour.



(c) True field at $t = 1.8$ hour.

(d) Estimated field at $t = 1.8$ hour.

Figure 25: True and estimated mean flow fields at $t = 0.2$ hour and $t = 1.8$ hour.

Table 9: Convergence conditions and quantitative results (mean) for parametric distributed MT simulations with time-varying flow at time $t = \{0.2, 1.8\}$ hour

Traj. tracing conv. condition	Estimation conv. condition	$\mathbf{e}^{rms}$ ($t = 0.2$ hour)	$\mathbf{e}^{rms}$ ($t = 1.8$ hour)
$\epsilon_\gamma = 10^{-2}$	$\epsilon_{\overline{\mathbf{f}}} = 30$	$x : 0.0115\, m/s$ $y : 0.0125\, m/s$	$x : 0.0043\, m/s$ $y : 0.0049\, m/s$

m/s in the y direction. The rms errors at 1.8 hour are 0.0043 m/s in the x direction and 0.0049 m/s in the y direction. These results are very close to the results of non-distributed MT presented in Section 5.1.3. These results show that the MT method for time-varying flow is successfully implemented in a distributed fashion.

CHAPTER VII

CONCLUDING REMARKS AND FUTURE DIRECTIONS

The main contribution of this thesis is a formulation of the problem of estimating an underlying flow field from trajectory information of underwater mobile sensing agents (UMSAs). The idea behind the problem is that the vehicle trajectory is affected by the underlying flow field and thus inherently provides information of the underlying flow field. The relationship between the underlying flow field and the vehicle motion is modeled as a nonlinear system of equations and computing the underlying flow field from the vehicle trajectory information is formulated as an inverse problem for this system of equations. The proposed method, referred to as *motion tomography* (MT), estimates the underlying flow field computationally faster and in higher resolution than existing oceanic flow field models by solving this inverse problem.

MT is first analyzed for time-invariant flow and flow field mapping through MT is achieved by iterating two key steps of trajectory tracing and flow field estimation. To address unknown nonlinear vehicle trajectories, trajectory tracing is incorporated into MT and its error bound is determined by the error in estimated heading and that in estimated flow field. Flow estimation algorithms are developed based on Kaczmarz-type methods that solve nonlinear systems of equations constructed for MT with non-parametric and parametric flow models. The validation and effectiveness of the method are demonstrated using simulation and experimental data. In the ideal case with simulation data, non-parametric MT show promising results but also show its inevitable dependency on vehicle trajectories. By incorporating a data-driven flow model that approximates the flow using basis functions with corresponding parameters, parametric MT generates less mapping error for a smooth field. The

experimental results demonstrate that the violations of assumptions (i.e., the constant vehicle speed and the time-invariant flow field) may degrade the mapping performance of MT, but flow field mapping through MT still provides valuable information in UMSA operation.

The MT method is then extended to time-varying flow fields. Temporal variability of flow can be incorporated into MT by discretizing the mapping domain in both space and time. To resolve the issue with the high dimension and sparsity of the solution variable arised by the spatio-temporal discretization of the domain, a data-driven flow model is incorporated. Then, MT constructs a time-varying flow field by estimating parameters of the flow model. Using parameters estimated through MT, a time-varying flow field is successfully constructed. Another approach incorporates both Eulerian data collected from stationary moorings and Lagrangian data collected from UMSAs into a data-driven flow model that formulates a generic environment model (GEM). Due to the coupling between temporal and spatial components in the flow model, estimation of temporal and spatial parameters for the GEM becomes a nonlinear filtering problem, which is decomposed into two linear sub-filtering problems in the proposed method. Each of the sub-filters estimates either temporal or spatial parameters by assimilateing Eulerian or Lagrangian data into the GEM, respectively. Assimilation of both Eulerian and Lagrangian data improves the GEM and also resolves the lack of the temporal variability in MT and the resulting GEM increases the navigation performance of UMSAs.

To account for the case in which vehicle trajectory information cannot be obtained collectively, distributed MT is developed so that individual vehicles can estimate a flow field with knowledge of the estimates shared among their neighbors only. This document has presented two distributed methods for solving a system of equations: a distributed Kaczmarz method for solving a linear system of equations and a

distributed Kaczmarz-type method for solving a nonlinear system of equations constructed for MT. The convergence analysis for both methods has been provided. The method for MT has been applied to distributed MT to account for the case in which vehicle trajectory information cannot be obtained collectively for MT. Distributed MT is validated through simulations and simulation results have demonstrated that the proposed method successfully estimates a map of a flow field in a distributed fashion.

In the future, a map of a flow field estimated by MT can be used to guide UMSAs in real time. As part of the effort towards this future work, the practical applicability of the method to realistic oceanographic features (e.g., eddies or filaments) will be explored in light of the characteristic time and length scales of the features relative to the speed, number, and placement of UMSAs. Future work will also address the performance bounds for MT in terms of the temporal and spatial variability of a flow field. Further improvements to MT for oceanographic application will also take advantage of scientific data collected by the UMSA along its trajectory that are not integrated along the trajectory but may have interdependencies with the underlying flow field. In addition to the guidance of UMSAs, the resulting flow field map from MT can be used to improve the ocean model through classical data assimilation processes. All these future directions, incorporating oceanographic and engineering principles, will significantly expand the method for more general application in the ocean sensing experiments.

REFERENCES

[1] Apte, A., Jones, C. K. R. T., Stuart, A. M., and Voss, J., "Data assimilation: Mathematical and statistical perspectives," *International Journal for Numerical Methods in Fluids*, vol. 56, pp. 1033–1046, mar 2008.

[2] Atkinson, L. P., Miller, J. L., Lee, T. N., and Dunstan, W. M., "Nutrients and chlorophyll at the shelf break off the southeastern United States during the Genesis of Atlantic Lows Experiment: Winter 1986," *Journal of Geophysical Research*, vol. 101, no. C9, pp. 20565–20578, 1996.

[3] Bishop, C. M., *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[4] Bleck, R., "An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates," *Ocean Modelling*, vol. 37, pp. 55–88, 2002.

[5] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[6] Censor, Y., "Row-action methods for huge and sparse systems and their applications," *SIAM Review*, vol. 23, no. 4, pp. 444–466, 1981.

[7] Chang, D., Liang, X., Wu, W., Edwards, C. R., and Zhang, F., "Real-time modeling of ocean currents for navigating underwater glider sensing networks," in *Cooperative Robots and Sensor Networks* (Koubâa, A. and Khelil, A., eds.), vol. 507 of *Studies in Computational Intelligence*, pp. 61–75, Springer Berlin Heidelberg, 2014.

[8] Chang, D., Wu, W., Edwards, C. R., and Zhang, F., "Motion tomography: Mapping flow fields using autonomous underwater vehicles," *International Journal of Robotics Research*, 2015 (under revision).

[9] Chang, D., Wu, W., and Zhang, F., "Glider CT: Analysis and experimental validation," in *Distributed Autonomous Robotic Systems* (Chong, N.-Y. and Cho, Y.-J., eds.), vol. 112 of *Springer Tracts in Advanced Robotics*, pp. 285–298, Springer Japan, 2016.

[10] Chang, D. and Zhang, F., "Distributed motion tomography for time-varying flow fields," in *OCEANS'16 MTS/IEEE Shanghai (OCEANS 2016)*, pp. 1–7, 2016.

[11] Chang, D. and Zhang, F., "Resolving temporal variations in data-driven flow models constructed by motion tomography," in *10th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2016)*, 2016 (accepted).

[12] Chang, D., Zhang, F., and Edwards, C. R., "Real-time guidance of underwater gliders assisted by predictive ocean models," *Journal of Atmospheric and Oceanic Technology*, vol. 32, no. 3, pp. 562–578, 2015.

[13] Chen, A. I., *Fast distributed first-order methods*. PhD thesis, Massachusetts Institute of Technology, 2012.

[14] Chen, A. I. and Ozdaglar, A., "A fast distributed proximal-gradient method," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012*, pp. 601–608, 2012.

[15] Chong, E. K. P. and Zak, S. H., *An introduction to optimization*. John Wiley & Sons, Inc., fourth ed., 2013.

[16] Cierniak, R., *X-Ray Computed Tomography in Biomedical Engineering*. Springer London, 2011.

[17] Claus, B., Bachmayer, R., and Williams, C. D., "Development of an auxiliary propulsion module for an autonomous underwater glider," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 224, no. 4, pp. 255–266, 2010.

[18] Crowley, M., Schofield, O., Glenn, S., and Whoriskey, F., "Gliderpalooza 2013 to modelpalooza 2014: Joint U.S. & Canadian ocean glider operations supporting multidisciplinary scientific research and education," in *2014 Oceans - St. John's*, pp. 1–5, 2014.

[19] Curtin, T. B. and Bellingham, J. G., "Progress toward autonomous ocean sampling networks," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 56, pp. 62–67, feb 2009.

[20] Curtin, T. B., Bellingham, J. G., Catipovic, J., and Webb, D., "Autonomous oceanographic sampling networks," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.

[21] Davis, R. E., Eriksen, C. C., and Jones, C. P., "Autonomous buoyancy-driven underwater gliders," in *Technology and Applications of Autonomous Underwater Vehicles* (Griffiths, G., ed.), pp. 37–58, CRC Press, 2002.

[22] Duchi, J. C., Agarwal, A., and Wainwright, M. J., "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[23] Dunbabin, M. and Marques, L., "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[24] Eichhorn, M., "Optimal routing strategies for autonomous underwater vehicles in time-varying environment," *Robotics and Autonomous Systems*, vol. 67, pp. 33–43, 2015.

[25] Fernández-Perdomo, E., Cabrera-Gámez, J., Hernández-Sosa, D., Isern-González, J., Domínguez-Brito, A. C., Redondo, A., Coca, J., Ramos, A. G., Fanjul, E. Á., and García, M., "Path planning for gliders using Regional Ocean Models: Application of Pinzón path planner with the ESEOAT model and the RU27 trans-Atlantic flight data," in *Proceedings of OCEANS 2010*, pp. 1–10, 2010.

[26] Fiorelli, E., Leonard, N. E., Bhatta, P., Paley, D. A., Bachmayer, R., and Fratantoni, D. M., "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 935–948, 2006.

[27] Fornberg, B. and Flyer, N., *A Primer on Radial Basis Functions with Applications to the Geosciences*. Society for Industrial and Applied Mathematics, 2015.

[28] Fratantoni, D. M. and Haddock, S. H., "Introduction to the autonomous ocean sampling network (AOSN-II) program," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 56, no. 3-5, p. 61, 2009.

[29] Girosi, F. and Poggio, T., "Networks and the best approximation property," *Biological Cybernetics*, vol. 63, no. 3, pp. 169–176, 1990.

[30] Gordon, R., Bender, R., and Herman, G. T., "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography," *Journal of theoretical biology*, vol. 29, no. 3, pp. 471–481, 1970.

[31] Gurgel, K.-W., Antonischki, G., Essen, H.-H., and Schlick, T., "Wellen Radar (WERA): A new ground-wave HF radar for ocean remote sensing," *Coastal Engineering*, vol. 37, no. 3-4, pp. 219–234, 1999.

[32] Hackbarth, A., Kreuzer, E., and Schröder, T., "CFD in the loop: Ensemble Kalman filtering with underwater mobile sensor networks," in *the ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, p. V002T08A063, 2014.

[33] Haidvogel, D. B., Arango, H., Budgell, W. P., Cornuelle, B. D., Curchitser, E., Di Lorenzo, E., Fennel, K., Geyer, W. R., Hermann, A. J., Lanerolle, L., Levin, J., McWilliams, J. C., Miller, A. J., Moore, A. M., Powell, T. M., Shchepetkin, A. F., Sherwood, C. R.,

Signell, R. P., Warner, J. C., and Wilkin, J., "Ocean forecasting in terrain-following coordinates: Formulation and skill assessment of the Regional Ocean Modeling System," *Journal of Computational Physics*, vol. 227, no. 7, pp. 3595–3624, 2008.

[34] Hart, P. E., Nilsson, N. J., and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[35] Heckman, C. R., Schwartz, I. B., and Hsieh, M. a., "Toward efficient navigation in uncertain gyre-like flows," *The International Journal of Robotics Research*, vol. 34, pp. 1590–1603, nov 2015.

[36] Herman, G. T., *Fundamentals of Computerized Tomography*. Advances in Pattern Recognition, London: Springer London, 2009.

[37] Horn, R. A. and Johnson, C. R., *Matrix Analysis*. Cambridge University Press, 1985.

[38] Isa, K., Arshad, M., and Ishak, S., "A hybrid-driven underwater glider model, hydrodynamics estimation, and an analysis of the motion control," *Ocean Engineering*, vol. 81, pp. 111–129, 2014.

[39] Jakovetic, D., Xavier, J., and Moura, J. M. F., "Fast Distributed Gradient Methods," *IEEE Transactions on Automatic Control*, vol. 59, pp. 1131–1146, may 2014.

[40] Jazwinski, A. H., *Stochastic processes and filtering theory*. Dover Publications, 1970.

[41] Kaczmarz, S., "Angenäherte auflösung von systemen linearer gleichungen," *Bulletin International de l'Academie Polonaise des Sciences et des Lettres*, vol. 35, pp. 355–357, 1937.

[42] Kaczmarz, S., "Approximate solution of systems of linear equations," *International Journal of Control*, vol. 57, no. 6, pp. 1269–1271, 1993.

[43] Kak, A. C. and Slaney, M., *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics, 2001.

[44] Kamath, G., Ramanan, P., and Song, W.-Z., "Distributed randomized Kaczmarz and applications to seismic imaging in sensor network," in *2015 International Conference on Distributed Computing in Sensor Systems*, pp. 169–178, 2015.

[45] Krener, A. J. and Ide, K., "Measures of unobservability," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 6401–6406, dec 2009.

[46] Kruger, D., Stolkin, R., Blum, A., and Briganti, J., "Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 4265–4270, 2007.

[47] Kumar, S., *Neural networks: A classroom approach.* Tata McGraw-Hill Education, 2004.

[48] Kuznetsov, L., Ide, K., and Jones, C. K. R. T., "A method for assimilation of Lagrangian data," *Monthly Weather Review*, vol. 131, no. 10, pp. 2247–2260, 2003.

[49] Lee, T. N., Yoder, J. A., and Atkinson, L. P., "Gulf Stream frontal eddy influence on productivity of the southeast U.S. continental shelf," *Journal of Geophysical Research*, vol. 96, no. C12, pp. 22191–22205, 1991.

[50] Leonard, N. E., Paley, D. A., Davis, R. E., Fratantoni, D. M., Lekien, F., and Zhang, F., "Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay," *Journal of Field Robotics*, vol. 27, no. 6, pp. 718–740, 2010.

[51] Lobel, I. and Ozdaglar, A., "Convergence analysis of distributed subgradient methods over random networks," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 353–360, 2008.

[52] Luettich, R. A., Westerink, J. J., and Scheffner, N. W., "ADCIRC: An advanced three-dimensional circulation model for shelves, coasts, and estuaries. Report 1. Theory and methodology of ADCIRC-2DDI and ADCIRC-3DL," tech. rep., Coastal Engineering Research Center, Vicksburg, Mississippi (U.S.), 1992.

[53] Lynch, K. M., Schwartz, I. B., Yang, P., and Freeman, R. A., "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[54] Martínez, J. M. and De Sampaio, R. J., "Parallel and sequential Kaczmarz methods for solving underdetermined nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 15, no. 3, pp. 311–321, 1986.

[55] Matei, I. and Baras, J. S., "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 754–771, 2011.

[56] Meldrum, D. T. and Haddrell, T., "GPS in autonomous underwater vehicles," in *Proceedings of the Sixth International Conference on Electronic Engineering in Oceanography*, pp. 11–17, 1994.

[57] Merckelbach, L. M., Briggs, R. D., Smeed, D. A., and Griffiths, G., "Current measurements from autonomous underwater gliders," in *Proceedings*

of the IEEE/OES/CMTC Ninth Working Conference on Current Measurement Technology, pp. 61–67, 2008.

[58] Meyn, K.-h., "Solution of underdetermined nonlinear equations by stationary iteration methods," *Numerische Mathematik*, vol. 42, no. 2, pp. 161–172, 1983.

[59] Mokhasi, P., Rempfer, D., and Kandala, S., "Predictive flow-field estimation," *Physica D: Nonlinear Phenomena*, vol. 238, no. 3, pp. 290–308, 2009.

[60] Mukhopadhyay, S., Wang, C., Patterson, M., Malisoff, M., and Zhang, F., "Collaborative autonomous surveys in marine environments affected by oil spills," in *Cooperative Robots and Sensor Networks* (Koubaa, A. and Khelil, A., eds.), vol. 554 of *Studies in Computational Intelligence*, pp. 87–113, Springer Berlin Heidelberg, 2014.

[61] Natterer, F., *The Mathematics of Computerized Tomography*. Society for Industrial and Applied Mathematics, 2001.

[62] Nedic, A. and Ozdaglar, A., "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[63] Nicholson, J. W. and Healey, A. J., "The present state of autonomous underwater vehicle (AUV) applications and technologies," *Marine Technology Society Journal*, vol. 42, no. 1, pp. 44–51, 2008.

[64] Paduan, J. D., Petruncio, E. T., Barrick, D. E., and Lipa, B. J., "Surface currents within and offshore of Monterey Bay as mapped by a multiple-site HF radar (CODAR) network," in *Proceedings of the IEEE Fifth Working Conference on Current Measurement*, pp. 137–142, IEEE, 1995.

[65] Park, J. and Sandberg, I. W., "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.

[66] Pawlowicz, R., Beardsley, B., and Lentz, S., "Classical tidal harmonic analysis including error estimates in MATLAB using T_TIDE," *Computers and Geosciences*, vol. 28, pp. 929–937, 2002.

[67] Penrose, R., "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, pp. 406–413, 1955.

[68] Pereira, A. A., Binney, J., Hollinger, G. A., and Sukhatme, G. S., "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.

[69] Petrich, J., Woolsey, C. A., and Stilwell, D. J., "Planar flow model identification for improved navigation of small AUVs," *Ocean Engineering*, vol. 36, no. 1, pp. 119–131, 2009.

[70] Popa, C. and Zdunek, R., "Kaczmarz extended algorithm for tomographic image reconstruction from limited-data," *Mathematics and Computers in Simulation*, vol. 65, no. 6, pp. 579–598, 2004.

[71] Prikopa, K. E., Straková, H., and Gansterer, W. N., "Analysis and comparison of truly distributed solvers for linear least squares problems on wireless sensor networks," in *Euro-Par 2014 Parallel Processing*, Lecture Notes in Computer Science, pp. 403–414, Springer International Publishing, 2014.

[72] Sánchez A, V., "On the number and the distribution of RBF centers," *Neurocomputing*, vol. 7, pp. 197–202, mar 1995.

[73] Schomberg, H., "An improved approach to reconstructive ultrasound tomography," *Journal of Physics D: Applied Physics*, vol. 11, no. 15, pp. L181–L185, 1978.

[74] Shay, L. K., Martinez-Pedraja, J., Cook, T. M., Haus, B. K., and Weisberg, R. H., "High-Frequency Radar Mapping of Surface Currents Using WERA," *Journal of Atmospheric and Oceanic Technology*, vol. 24, pp. 484–503, mar 2007.

[75] Shchepetkin, A. F. and McWilliams, J. C., "The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[76] Shi, L., Song, W.-Z., Kamath, G., Xing, G., and Liu, X., "Distributed least-squares iterative methods in networks : A survey," *Submitted to Computing Journal*, 2013.

[77] Smith, R. N., Chao, Y., Li, P. P., Caron, D. A., Jones, B. H., and Sukhatme, G. S., "Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model," *The International Journal of Robotics Research*, vol. 29, no. 12, pp. 1475–1497, 2010.

[78] Smith, R. N., Pereira, A. A., Chao, Y., Li, P. P., Caron, D. A., Jones, B. H., and Sukhatme, G. S., "Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 4770–4777, 2010.

[79] Smith, R. N., Schwager, M., Smith, S. L., Jones, B. H., Rus, D., and Sukhatme, G. S., "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, pp. 714–741, sep 2011.

[80] Soulignac, M., "Feasible and optimal path planning in strong current fields," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 89–98, 2011.

[81] Strohmer, T. and Vershynin, R., "A randomized Kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, pp. 262–278, 2008.

[82] Sundhar Ram, S., Nedić, A., and Veeravalli, V. V., "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[83] Szwaykowska, K. and Zhang, F., "Trend and bounds for error growth in controlled Lagrangian particle tracking," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 10–25, 2014.

[84] Tanabe, K., "Projection method for solving a singular system of linear equations and its applications," *Numerische Mathematik*, vol. 17, no. 3, pp. 203–214, 1971.

[85] Techy, L., "Optimal navigation in a planar time-varying point-symmetric flowfield," in *IEEE Conference on Decision and Control and European Control Conference*, pp. 7325–7330, IEEE, dec 2011.

[86] Wang, S. X., Sun, X. J., Wu, J. G., Wang, X. M., and Zhang, H. W., "Motion characteristic analysis of a hybrid-driven underwater glider," in *Proceedings of OCEANS 2010 IEEE Sydney*, pp. 1–9, 2010.

[87] Witt, J. and Dunbabin, M., "Go with the flow: Optimal AUV path planning in coastal environments," in *Proceedings of the 2008 Australian Conference on Robotics and Automation*, 2008.

[88] Wu, W., Chang, D., and Zhang, F., "Glider CT: Reconstructing flow fields from predicted motion of underwater gliders," in *Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems*, p. 47, 2013.

[89] Wu, Y., Wang, H., Zhang, B., and Du, K.-L., "Using Radial Basis Function Networks for Function Approximation and Classification," *ISRN Applied Mathematics*, vol. 2012, pp. 1–34, 2012.

[90] Yuh, J. and West, M., "Underwater robotics," *Advanced Robotics*, vol. 15, no. 5, pp. 609–639, 2001.

[91] Zhang, F., Marani, G., Smith, R. N., and Choi, H. T., "Future Trends in Marine Robotics [TC Spotlight]," *IEEE Robotics and Automation Magazine*, vol. 22, no. 1, pp. 14–122, 2015.

[92] Zhao, L., Song, W.-Z., Shi, L., and Ye, X., "Decentralised seismic tomography computing in cyber-physical sensor systems," *Cyber-Physical Systems*, pp. 1–22, 2015.

VITA

Dongsik Chang received his B.S. degree in 2007 from Hanyang University in Seoul, Korea. He began attending Georgia Tech in 2008 and joined the Georgia Tech Systems Research group, a research group directed by Dr. Fumin Zhang, in 2009. In December 2010, he completed his M.S. degree in Electrical and Computer Engineering and is currently a Ph.D. candidate under the advisement of Dr. Fumin Zhang. His current research work focuses on a tomographic approach to environmental sensing and modeling using mobile sensor networks for their guidance. His research interests include robotics, mobile sensor networks, path planning, control theory, and stochastic systems.